Crash−Kurs "Linux", Version 3.9

Trotz Sorgfalt bei der Zusammenstellung dieses Manuskripts kann ich nicht dessen Richtigkeit und Brauchbarkeit garantieren. Daher

BENUTZUNG DES SKRIPTS AUF EIGENE GEFAHR.

Insbesondere ist es bewußt UNVOLLSTÄNDIG, um eine möglichst knappe Darstellung zu erzielen.
Das Ausprobieren der enthaltenen Beispiele kann Ihre Daten unwiederbringlich ZERSTÖREN und/oder Hardware irreperabel BESCHÄDIGEN. Ich übernehme dafür keine Haftung.

Inhaltsverzeichnis

2. Über den Crash−Kurs

2.1 Zielgruppe

Die vorliegenden Seiten waren in erster Linie Unterlagen zu meinen 1− und 5−tägigen Einführungskursen zu Linux gedacht, die ich früher zweimal pro Jahr an der Universität zu Köln hielt. Ohne den zugehörigen Vortrag muß man diese Unterlagen selbst zum Leben erwecken, indem man alle Beispiele und Bemerkungen ausprobiert.

Der Inhalt orientiert sich an dem, was ich in den vergangenen 5 Jahren Anfängern und Interessierten zum Thema Linux erzählt habe. Ich habe versucht, die "Dauerbrenner" in einer möglichst logischen Reihenfolge zu präsentieren und trotz der extrem knappen Darstellung eine gewisse Verständlichkeit zu gewährleisten.

Allzu ausführliche Texte sind nach meiner Auffassung ein Indiz für Unzulänglichkeiten, die mit hohem Zeitaufwand beschrieben und somit zementiert werden, anstatt sie mit einigem Programmieraufwand ein für alle mal zu beheben. Der "midnight commander" (siehe 6.3), KDE und GNOME sind in dieser Hinsicht positive Beispiele.

Mittlerweile (November 1997) ist das Manuskript etwas angestaubt, aber mir fehlt die Zeit für eine Generalüberholung. Stattdessen habe ich mich entschieden, im Addison−Wesley Verlag das Buch "Intranetworking mit Linux" herauszubringen. Darin werden die hier etwas vernachlässigten letzten Kapitel ausführlicher dargestellt.

2.2 Notation

- o Beispiele, die wortwörtlich auf der Kommandozeile ausprobiert werden können, sind eindeutig mit den Zeichen » und « eingegrenzt. Dies scheint mir notwendig, da alle anderen Anführungszeichen unter Linux spezielle Bedeutung haben.
  Programmnamen sind ebenfalls mit diesen Zeichen gekennzeichnet.

- o Buchstaben zwischen < und > symbolisieren eine Taste auf der Tastatur des Computers.

- o Beispiele, die sinngemäß ausprobiert werden sollen, sind lediglich mit normalen Anführungszeichen " und " gekennzeichnet, sie gehören nur in diesem Fall nicht zum Beispiel.

- o Datei− und Verzeichnisnamen sind zwischen den Apostroph−Zeichen ‘ und ’ eingeschlossen.

- o URL−Adressen (siehe 41.1) werden ohne besondere Markierungen zitiert, erhalten dafür aber eine eigene Zeile.

2.3 Nutzungslizenz

Dieses Manuskript darf zu privaten Zwecken nur dann weitergegeben werden, wenn mindestens die ersten beiden Seiten in unveränderter Form weitergegeben werden. Änderungen am Original müssen vor der Weitergabe eindeutig als solche gekennzeichnet werden.

Kommerzielle Nutzung (insbesondere der Vertrieb auf CD−ROM) ist nur erlaubt, wenn alle der folgenden Bedingungen erfüllt sind:

- Dieses Manuskript muß unverändert und vollständig weitergegeben werden.
- Es dürfen höchstens anteilige Kosten für den Datenträger (Papier, CD−ROM Rohling) berechnet werden.
- Es darf keine Werbung mit meinem Namen oder für dieses Manuskript gemacht werden.

2.4 Erhältlichkeit

Die neuste Version dieses Manuskript ist stets unter den URLs

http://www.shop.de/~winni/linux/

verfügbar.

2.5 Voraussetzungen

Ausgangspunkt des Manuskripts ist ein ist lauffähiges Linux−System, auf dem die Kommandozeile benutzt werden kann.

3. Allgemeines zu Linux

Glossar:
- o Mit "Server" werden u.a. Computer bezeichnet, die in einem Netz (aus Computern) Daten und Dienstleistungen zur Verfügung stellen.
- o "POSIX" bezeichnet einen Programmierstandard in der Unix−Welt.

3.1 Was ist Linux?

- o "Linux" ist genaugenommen ein Betriebssystemkern (engl. "kernel"), dessen Entwicklung im April 1991 vom finnischen Studenten Linus Torvalds gestartet wurde.
- o Der Name geht auf "Linus' Unix" zurück und zeigt damit an, daß Linux ein unix−artiger Betriebssystemkern ist.
- o Dieser Betriebssystemkern ist nach dem Start des Computers die oberste Instanz und hat im wesentlichen folgende Fähigkeiten:
  - − "multiuser": jede Personen, die den Computer benutzt, hat eine eigene "Privatsphäre" (geschützt durch ein Geheimwort)
  - − "multitasking": mehrere Aufgaben (engl. "tasks") können quasi gleichzeitig erledigt werden (siehe 26.2)
  - − die Ressourcen des Computers (wie Arbeitsspeicher [RAM], Rechenleistung und Netzwerk) werden den konkurrierenden Programmen zugeteilt (d.h. keine Selbstbedienung), dadurch wird eine gewisse Sicherheit erreicht
  - − verschiedene Modelle von Computerbestandteilen mit gleicher Funktion (z.B. verschiedene Soundkarten) können von den Programmen in stets gleicher Art und Weise angesprochen werden; eine große Hilfe für Programmierer!
- o Der Betriebssystemkern von Linux ist in der Programmiersprache C geschrieben und liegt im Quelltext vor.
- o Ohne zusätzliche Programme (Software) ist ein Betriebssystemkern ziemlich nutzlos.

3.2 Linux − das Betriebssystem

- o Um die Fähigkeiten des Betriebssystemkerns komfortabel und sinnvoll nutzen zu können, benötigt man mindestens:
  - − Programme zur Erkennung und Behebung von Fehlern
  - − den Zugriffsschutz, d.h. die Abfrage des Geheimwortes
  - − eine Befehlszeile zum Start weiterer Programme
  - − ein System zum automatischen Start der oben genannten Programme
- o Dieses absolut notwendigen Komponenten bezeichnet man landläufig als "Betriebssystem" (eine genauere Definition lasse ich aus).
- o Die Installation des Linux−Betriebssystems ist weitgehend automatisiert und menügeführt.

3.3 Linux−Distributionen

- o "Linux−Distributionen" sind kinderleicht zu installierende Sammlungen von Programmen "für Linux"; sie enthalten außerdem das Betriebssystem.
- o Linux−Distributionen werden von verschiedenen Interessengruppen in leicht unterschiedlicher und wechselnder Zusammensetzung, bzw. Qualität angeboten.
- o Es gibt keine "offizielle" Linux−Distribution und somit läßt sich von Linux nicht sprechen wie von DOS, Win oder OS:2.
- o Typischerweise befinden sich nach der Installation mehr als 1000 Programme auf der Festplatte des Computers:
  - − eine netzwerkfähige, grafische Oberfläche (X11, XFree86)
  - − Programmiersprachen (C, C++, phyton, perl, basic, ...)
  - − Internet−Browser (netscape, mosaic, arena)
  - − Internet−Server (WWW, FTP, smb, e−mail, ...)
  - − Text−Editoren
  - − das Text−Satzsystem "TeX"

  ... jeweils mit fast unüberschaubar viel Zubehör und mehr Dokumentation als man lesen kann.

3.4 Woher stammt die ganze Software?

- o Programmieren ist ebenso wie das Malen oder Nähen eine kreative Tätigkeit, die Freude bereitet.
- o Die meiste Software ist nicht speziell für Linux geschrieben worden (sondern der Betriebssystemkern wurde so geschrieben, daß die meiste Software auf einem PC unter Linux läuft ...).
- o Das Internet und das Amateurradio schaffen die Infrastruktur zur sekundenschnellen Kommunikation der Software−Entwickler; erst dadurch wurden so große Projekte wie Linux überhaupt möglich.
- o Jedoch ist nicht alle Software für Linux kostenlos erhältlich, in letzter Zeit gibt es erfreulicherweise immer mehr Firmen, die ihre Produkte auch für Linux anbieten.

3.5 Zusammenfassung

- o Im allgemeinen Sprachgebrauch wird nicht zwischen dem Betriebssystemkern, dem Betriebssytem oder den Linux−Distributionen unterschieden, was manchmal für Verwirrung sorgt ...
- o Linux wird von einer Fangemeinde aus Programmierern entwickelt, die Linux selbst einsetzen; daher werden Fehler schnell erkannt und behoben; die Entwicklung ist sehr rasant.
- o Linux ist kostenlos, zuverlässig, schnell und POSIX−konform.
- o Technisch gesehen ist Linux eine gut gelungene Umsetzung von (mehr oder weniger) bekannten Konzepten (und damit veraltet ?).

4. Weitere Details zu Linux

4.1 Konkret: was man mit Linux anstellen kann

- Workstation
  Linux verwandelt einen PC in eine leistungsfähige Workstation. Als Arbeitsumgebung zum Verfassen von TeX−Dokumenten und als Entwicklungsumgebung zum Schreiben von Software hervorragend geeignet. Beispiele: »xwpe« (ähnlich der Oberfläche von TurboC), »emacs« (AucTeX−Modus)

- Kostengünstiges X−Terminal
  Installiert man nur das Grundsystem (OHNE C−Compiler, TeX, usw.) plus XFree86, so erhält man automatisch ein X−Terminal zu einem unschlagbarem Preis.

- Parallelrechner
  PC−Pools lassen sich per Boot−Diskette kurzfristig und ohne Änderungen an den bestehenden Betriebssystemen in einen Parallelrechner verwandeln (Infos bei Andreas Erdmann <erdmann@zpr.uni−koeln.de>).
  Die Rechenleistung einiger Dutzend zusammengeschalteter Pentium−PCs ist beachtlich.

- WWW−Server
  Mittlerweile gibt es ein reichhaltiges Angebot an Programmen, mit denen sich WWW−Server unter Linux realisieren lassen. Der "Apache" ist der meistgebrauchte Server; mein persönlicher Favorit ist der "Roxen".

- Allgemeine Server−Funktionen
  Nahezu jede Server−Funktion läßt sich mit Linux realisieren. Sei es als File−Server für DOS/Win−PCs, als e−mail HUB, FTP−Server, Firewall ...

4.2 Nachteile von Linux

- Die Fähigkeit zum Verwalten mehrerer Benutzer bedeutet einen höheren Verwaltungsaufwand (in jeder Hinsicht).
- Hinter Linux und den meisten seiner Programme steht ein Konzept, ohne dessen Kenntnis man hilflos ist.
- Linux ist nicht besser oder schlechter als die herkömmlichen Betriebssysteme für den PC, sondern hat eine andere Zielgruppe; deswegen lohnen sich der Aufwand für die Installation und die Einarbeitungszeit nicht in jedem Fall.

4.3 Welche Hardwarevoraussetzungen benötigt Linux/i386?

Theoretisches Minimum:	386 SX, 2MB RAM, 5 MB Festplatte
Akzeptables Minimum:	386 DX/40 mit Coprozessor (!), 16 MB RAM
Akzeptables System:	486 DX/66, 32MB RAM, 600 MB Festplatte
Sehr schnelles System:	P200, 64MB RAM, 4GB Festplatte

Weitere Plattformen: Atari, Amiga, MAC, DEC−Alpha, Sparc, MIPS, Strong Arm

4.4 Rechtlicher Status

- Da Linux aus sehr vielen Komponenten mit unterschiedlichem Copyright besteht, läßt sich keine einheitliche Aussage machen.
- Üblicherweise behalten sich die Autoren die Rechte an ihren Werken vor, erlauben aber die Nutzung unter der Auflage der "General Public Licence" (GPL), die im wesentlichen aussagt:
  - die Software DARF weitergegeben werden, aber nur unter denselben Bedingungen
  - die Software kommt ohne irgendwelche Garantien, eine Haftung ist ausgeschlossen
  - es dürfen bei Verkauf lediglich Kosten für den Vertrieb berechnet werden (CD−ROM, Disketten, Papier, Verpackung)
  - es werden KEINE Lizenz−, Registrations− oder Benutzungs−gebühren fällig
  - insbesondere müssen stets die Quelltexte verfügbar sein

- Die GPL wurde von der "Free Software Foundation" (FSF) entworfen und dient als Copyright für deren GNU−Programme (»gcc«, »emacs«, ...); manchmal wird sie auch als "Copyleft" bezeichnet.

- Nicht alle Linux−Programme unterliegen der GPL, daneben gibt es noch BSD−artige Copyrights, das Copyright des MIT oder die "Artistic Licence" von »perl«.
- "Freeware" und "Shareware" haben als Copyright fast keine Bedeutung unter Linux.

4.5 Erhältlichkeit

- Linux ist erhältlich
  - auf CD−ROM
  - im Internet via FTP, NFS oder AFS

- In den Anfangstagen war Linux auch auf Disketten erhältlich; mittlerweile hat Linux einen solchen Umfang, daß ca. 100 Disketten notwendig wären ...

4.6 Geschichtliches

- 1991 kauft sich der finnische Student Linus Torvalds einen 386er PC (auf Ratenzahlung ?), um mit den Möglichkeiten des 80386 Prozessors zu experimentieren.
- Aus den ersten Versuchen wird schnell ein einfacher, aber brauchbarer multitasking−Betriebssystemkern.
- Zunächst wurde Linux unter Minix, einem Lehrbetriebssystem von A. Tannenbaum ("Linux is obsolete"), entwickelt.
- Nachdem der C−Compiler »gcc« und weitere wichtige Entwicklungs−werkzeuge unter Linux liefen, konnte Linux unter Linux weiter−entwickelt werden.
- Der Betriebssystemkern wurde von Grund auf neu programmiert.
- Die Weiterentwicklung fand und findet unter Beteiligung von vielen interessierten Programmierern im Internet statt; Linus ist und war nie der einzige Entwickler.

5. Rüstzeug und generelle Bemerkungen

Es ist nicht unbedingt notwendig, sich jeden Punkt vor dem Weiterlesen zu merken; man muß im Zweifelsfalle nur wissen, daß er hier steht.

5.1 Allgemeine Regeln für den Umgang mit Linux

- o Alle Befehle werden ohne Nachfrage ausgeführt, also Vorsicht.
- o Regelmässig Sicherheitskopien anlegen.
- o Befehle zur Systemverwaltung sind root (siehe 16.5) vorbehalten.
- o Es wird i.d.R. zwischen Gross− und Kleinschreibung unterschieden.

5.2 Gebräuchliche Abkürzungen für Tastenkombinationen

- o Wird eine beliebige Taste gleichzeitig mit einer der folgenden Tasten gedrückt, so spricht man von einer Tastenkombination.

```
Beschriftung    Abkürzungen     Bedeutung
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
_/\_            shift−   S−     Umschaltung auf Großbuchstaben;
                                engl. "shift"
Strg (ctrl)     ^        C−     "Steuerung", "control"
Alt             Alt−     A−     alternative Belegung
|<− −>|         Tab−            Tabulatortaste (Einrückung)
<>  (Diamant)   Meta−    M−     übergeordnete Belegung
´ ´ (Lücke)     SPC             Leerzeichen, engl. "space"
Esc             ESC             "Escape"−Taste
<−−|            BS              Backspace (linkes Zeichen löschen)
```

- o Wird die <Alt>−Taste nicht erkannt, so kann man diese durch Vorausschicken von <ESC> simulieren (NICHT gleichzeitig drücken). Nützlich bei Arbeiten via Modem oder via »telnet«.
- o Die Funktion der <Meta>−Taste wird oft von der <Alt>−Taste übernommen.
- o Die Tastenkomination werden abgekürzt durch das Voranstellen der Tastenbeschriftungen oder der Kürzel.

```
        Bsp.:   <Strg−c> (drücke Tasten <Strg> und <c> gleichzeitig)
                <C−c> ist äquivalent zu <Strg−c>
                <Strg−Alt−Entf> rebootet Linux (also Vorsicht!)
```

5.3 Verbreitete Funktionen einiger Tasten(−kombinationen)

```
<q>     <Esc>   "quit", "escape", Programm verlassen
<h>     <?>     "help", Hilfestellung anzeigen
</>     <s>     Text durchsuchen
<SPC>           eine Seite weiterblättern

<Strg−c>        Programm beenden
<Strg−d>        "end of file" (EOF), Eingabe beenden
<Strg−q>        Ausgabe fortsetzen (Anhalten mit <Strg−s>)
<Strg−l>        Bildschirm neu zeichnen (auffrischen)
```

Falls einmal die Angaben in diesem Manuskript nicht auf den eigenen Dateibaum zutreffen: DON’T PANIC. Man kann sich behelfen:

5.4 Aufspüren von Dateien und Wörtern

»find /etc −name "xinitrc" −type f −print«
: sucht nach der Datei ‘xinitrc’ im ‘/etc’−Ast und zeigt alle Vorkommen mit dem kompletten Pfadnamen an; (’−type f’ um nur nach Dateien [engl. "files"] zu suchen).

»locate xinitrc«
: wie »find«, greift jedoch auf eine find−Datenbank zurück, wodurch die Suche extrem beschleunigt wird. Die notwendige Datenbank wird (als Systemverwalterin root) mit dem Befehl »su −l nobody −c updatedb« erzeugt. Nicht auf allen Systemen vorhanden, aber unbedingt vorzuziehen!

»grep −i loadkeys /etc/rc.boot/*«
: zeigt die Dateien im Verzeichnis ‘/etc/rc.boot’ an, in denen der Befehl »loadkeys« ausgeführt wird (’−i’ zum Ignorieren der Gross−/Kleinschreibung).

»grep −i loadkeys ‘find /etc −name "*" −type f −print‘«
: ist eine Kombination von »find« und »grep«.

»strace −eopen,stat −f −o /tmp/dump emacs«
: ruft den »emacs« auf und protokolliert (’−o’) in der Datei ‘/tmp/dump’ die Zugriffe (’−eopen,stat’) auf (Konfigurations−)Dateien. Außerdem werden alle Kind−Prozesse angezeigt (’−f’) [siehe 26.4].

5.5 Suchmuster

- o Suchmuster dienen zur Formulierung dessen, was man sucht.
  Bsp.:   ‘search*’ (alle Wörter, die mit ’search’ beginnen)

- o Es gibt hauptsächlich zwei Arten von Suchmustern:
  − "stringsearch"   (einfache Wortsuche)
  − "regular expressions"   (reguläre Ausdrücke)
- o Beide sind nicht 100%ig standardisiert.

- o "stringsearch" erlaubt meist nur eine Suche nach Teilworten; ein Stern steht für beliebig viele "fehlende" Zeichen.
  Bsp.:   ‘search*’   oder   ‘*.txt’

- o "regular expressions" erlauben sehr komplexe Suchmuster und sind unter Unix weit verbreitet.
- o Obgiges Beispiel sieht mit "regular expressions" so aus:
  Bsp.:   ‘search.*’   oder   ‘.*\.txt’

- o Der Punkt steht für ein beliebiges Zeichen, der Stern für "vorheriges Zeichen beliebig oft".
- o Ein Hut ´^´ steht für "Anfang der Zeile", ´$´ für das Ende.
- o Dokumentation zu regulären Ausdrücken findet man mit den Befehlen »man grep«, »man regexp« (Programmiersprache TCL), »man sed«.

6. Erste Schritte mit Linux

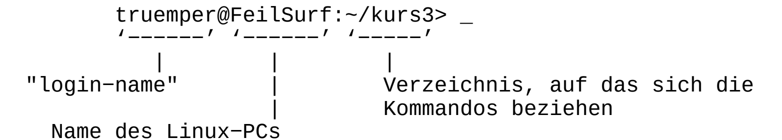
Die folgenden Ausführungen findet man auf den hinteren Seiten nochmals, und zwar mit mehr Details, als an dieser Stelle angebracht wären.

6.1 Das login–Prompt

- o Der Zugang zu einem Linux–PC ist durch einen Namen ("login") und ein Geheimwort ("password") geschützt.
- o Hat man beides eingegeben, so startet Linux eine Kommandozeile, auf der man arbeiten kann.

6.2 Die Kommandozeile

- o Ist man auf der Kommandozeile angelangt, so findet man einen ein schwarzen Block oder einen nervös blinkenden Unterstrich vor; beide werden als "Cursor" bezeichnet und markieren die Stelle, an der die eingetippten Zeichen erscheinen.
- o Die Kommandozeile beginnt etwa so:



- o Den Bereich vom Anfang der Zeile bis zum Cursor bezeichnet man als "Prompt". Man kann es selbst definieren, z.B.:

  »PS1='\u@\h:\w> '«

  (´\u´= "username", ´\h´= "hostname", ´\w´= "working directory")

- o Die Kommandozeile wird aus historischen Gründen als "shell" ("Muschel") bezeichnet, diese Bezeichnung wird im folgenden verwendet.
- o Die shell dient zum Aufruf der Programme (siehe 26.1), mit denen man eigentlich arbeiten will. Ein typischer Programmaufruf sieht etwa so aus:

```
truemper@FeilSurf:~/kurs3>  less  –MMI  vortrag3.txt
‘–––––––– Prompt ––––––––’  ‘––’  ‘––’  ‘–– Datei –’
                             |     |
                  Programmname     Optionen
```

6.3 Alternativen zur Kommandozeile

- o Für Anfänger ist der »mc« ("midnight commander") eine große Hilfe; er ersetzt die Kenntnis vieler Linux–Kommandos.

6.4 Optionen

- o "Optionen" beginnen meist mit einem ´–´ und beeinflussen die Grundfunktionen des Programms. Man gibt sie auf der Kommandozeile hinter dem Programmnamen an.

  Bsp.: »less –MMI«

- o Zwei Formate zur Angabe von Optionen sind verbreitet:

  - herkömmliche Kurzform: '–Option'
    Bsp.: »less –?« für das Anzeigen der Hilfe

  - GNU–Format: '––Option'
    Bsp.: »less ––help« für das Anzeigen der Hilfe

- o Die Reihenfolge kann entscheidend sein (z.B. beim C–Compiler).
- o ´ –– ´ (Doppel–Minus alleine) bedeutet "jetzt folgen keine Optionen mehr"
  Bsp.: »open –s –– bash –login« (siehe 17.2)

6.5 Argumente und Parameter

- o "Argumente" und "Parameter" sind Werte oder (Datei–)Namen, die Programmen mitgeteilt werden sollen.

  Bsp.: »less vortrag3.txt«

- o ´ – ´ (Minus) anstelle eines Dateinamens bedeutet "die Daten werden über den Ein– oder Ausgabekanal gelenkt" (siehe 22.5).
- o Leer– und Sonderzeichen in Parametern müssen gequotet werden.

6.6 Quoting (dt.: "zitieren")

- o Unter "quoting" versteht man die Befreiung eines Zeichens von einer speziellen Bedeutung (nicht nur auf die Kommandozeile bezogen).
- o "Das Zeichen ; trennt Sätze; ein Strich ...": Menschen können aus dem Zusammenhang erkennen, daß der erste Strichpunkt wörtlich zu verstehen ist und nicht teil der Grammatik ist ... dem Computer muß dies mitgeteilt werden!

- o Der Backslash (´\´) wirkt nur auf das unmittelbar nachfolgende Zeichen [insbesondere dient er NICHT zum Aufbau von Verzeichnisnamen wie unter DOS].
  Bsp.: »echo Das Zeichen \; trennt Sätze\; ...«

- o Anführungszeichen schützen Leer– und Sonderzeichen, die nicht der Ersetzung unterliegen.
  Bsp.: »echo "Das Zeichen ; trennt Sätze; ..."«
  »echo "Hallo Welt; Hallo $LOGNAME"«

- o Apostroph–Zeichen schützen alles.
  Bsp.: »echo 'Hallo Welt; Hallo $LOGNAME'«

7. Grundzüge der Dateiwaltung von Linux

7.1 Dateien (engl. "files")

- Dateien sind die Träger der Information (Texte, Bilder, usw.)
- Zulässige Dateinamen:

Länge:	1–255 Zeichen
Format:	kein spez. Format erforderlich (z.B. "DOS 8+3")
erlaubte Zeichen:	´/´ ist kein zulässiges Zeichen für Datei– und Verzeichnisnamennamen; die Beschränkung auf ´a–z,A–Z,0–9,.,_,+,–´ ist nicht notwendig, aber sinnvoll.

7.2 Elementare Kommandos zur Handhabung von Dateien

Aktion	Prgrogramm		Beispiele
auflisten	»ls«	("list")	»ls« »ls –l« »ls –la $HOME | more«
ausgeben	»cat«	("concatenate")	»cat /proc/cpuinfo«
kopieren	»cp«	("copy")	»cp /etc/profile .«
verschieben	»mv«	("move")	»mv profile /tmp«
löschen	»rm«	("remove")	»rm -i /tmp/profile«
klassifizieren	»file«		»file /etc/magic«
vergleichen	»diff«, »cmp«		siehe 12.3

7.3 Verzeichnisse (engl: "directories")

- Benutzersicht: Verzeichnisse sind hierarchisch organisierte Ablagefächer für Dateien; sie können selbst wieder Verzeichnisse enthalten. Zulässige Namen: siehe 7.1
- Den Weg zu einem in der Hierarchie tieferliegendem Verzeichnis wird mit ´/´ gebildet.
  Bsp.: ‘/usr/local/bin/’
- Besondere Verzeichnisse:

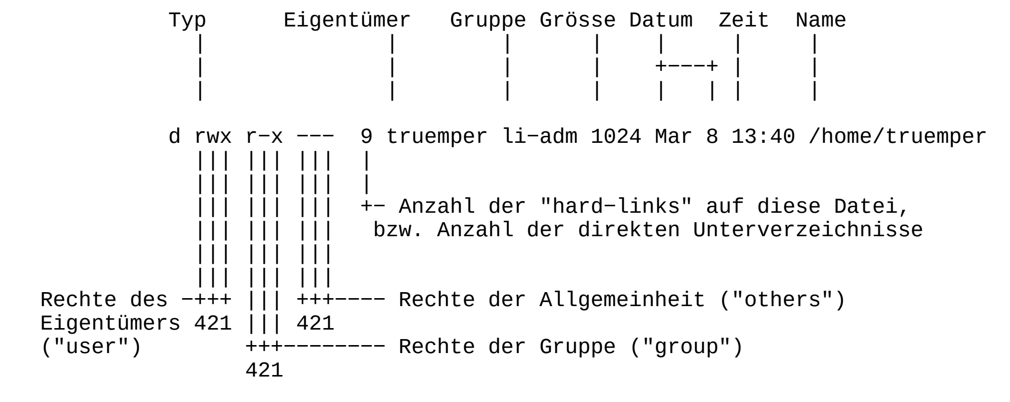
.	Verzeichnis, in dem man sich gerade befindet
..	übergeordnetes Verzeichnis
~	das eigene, private Verzeichnis
/	oberstes Verzeichnis der Hierarchie (siehe 31.1)

7.4 Umgang mit Verzeichnissen

Aktion	Prg.		Beispiele
erzeugen	»mkdir«	("make directory")	»mkdir /tmp/verz«
wechseln:	»cd«	("change directory")	»cd /« »cd –«, »cd +«
	»pushd«		»pushd /tmp«
	»popd«		»popd«
anzeigen	»ls«	("list")	»ls –la« (list long)
verschieben	»mv«	("move")	
löschen	»rmdir«	("remove directory")	»rmdir /tmp/verz«
	»rm –rf« (ohne Nachfrage und rückstandlos)		

7.5 Standard Zugriffs– und Ausführungsrechte

Abb. 1: Erläuterung der Ausgabe von »ls –ld ~«



Im obigen Beispiel können nur der Eigentümer "truemper" und Mitglieder der Gruppe "li-adm" in das Verzeichnis ‘/home/truemper’ wechseln (´x´) und dort auf Dateien und das Inhaltsverzeichnis zugreifen (´r´).
Außerdem kann nur der Eigentümer neue Dateien erzeugen oder alte löschen (´w´).

Kürzel	Recht	Erlaubt bei Dateien ...	Erlaubt bei Verzeichnissen ...
–		nichts	nichts
r	read	Ansehen der Datei (Leserecht)	Ansehen des Inhalts eines Verzeichnis
w	write	Verändern der Datei (Schreibrecht)	Löschen aller Dateien, Erzeugen von Dateien
t		keine Wirkung	nur eigene Dateien können gelöscht werden (‘/tmp’)
x	execute	Ausführen einer Datei als Programm	Betreten des Verzeichnisses
s		siehe 27.6	Siehe »man 5 mount«, Option ´grpid´

- Zugriffsrechte ändern: »chmod« ("change mode")
  Bsp.: »chmod u=rwx,go=rx ~« (u=user,g=group,o=others)
  »chmod 755 ~« (7=4+2+1, 5=4+1)
- Zugriffsrechte für "zukünftige" Dateien festlegen: »umask«
  Bsp.: »umask u=rwx,g=rx,o= « (evtl. nur vorübergehend
  »umask –S« wirksam; siehe 22.2)

Kürzel	Dateityp	Kürzel	Dateityp
–	Datei (engl.: "file")	b	block–device
d	Verzeichnis ("directory")	c	character– "
l	Verweis (engl.: "link")	s	socket
		p	pipe

8. Rund um Dateien

8.1 Kopieren von Dateien

- Der Befehl »cp« kopiert Dateien innerhalb des Linux-Dateibaums. Ganze Verzeichnisse können mit der Option '-a' kopiert werden.
  Bsp.: »cp -a /etc/skel /tmp«

- Auf fremde Maschinen innerhalb eines Computernetzwerks (z.B. Internet) kopieren: »scp« ("remote-copy", siehe 41.4).
  Bsp.: »scp localhost:/etc/profile . «

- Mit »tar« kann man Archive und Backups erzeugen (nicht beschränkt auf Bänder).
  Bsp.: »tar -Mczf /dev/fd0 ~/mail«

  In diesem Beispiel wird das Verzeichnis '~mail' auf Disketten ('-M' für mehrere) gesichert und dabei komprimiert ('z').

- "1:1"-Kopien von ganzen Datenträgern oder teilweise Kopien von Dateien können mit »dd« erzeugt werden.
  Bsp.: »dd if=/dev/fd0 of=Diskette_7«

8.2 Umgang mit DOS-Dateien

- Der Oberbegriff für alle Werkzeuge zum Umgang mit DOS-Dateien ist »mtools«.
  Doku.: »man mtools«

- Die Zuordnung von Laufwerksbuchstaben zu Geräten (Floppy, Festplatte) wird in der Datei '/etc/mtools.ref' konfiguriert.
  Bsp.: »mdir a:«

- Datenträger im DOS-Format können auch dauerhaft in den Dateibaum von Linux integriert werden (siehe 32.1).
  Bsp.: »mkdir /floppy;\
  mount -t msdos /dev/fd0 /floppy«

8.3 Format der Textdateien

- Die Textdateien der verschiedenen Betriebssysteme unterscheiden sich im Zeichensatz und dem Zeilenende.

	Zeichensatz	Zeilenende	
Linux	ISOLATIN-1	LF	(= "linefeed", Zeilenvorschub [neue Zeile], <Strg-J>)
MAC		CR	(= "carriage return", Wagenrücklauf [Anfang der Zeile], <Strg-M>)
DOS	IBMPC	CR LF	

- Umwandlung mit »dos2unix«, bzw. »unix2dos« oder »recode ibmpc:lat1« bzw. »recode lat1:ibmpc«.
  Bsp.: »recode lat1:ibmpc vortrag.txt«

8.4 Gerätedateien

- Hardware wird theoretisch über spezielle Dateien im Verzeichnis '/dev' angesprochen, siehe 29.1 und 30.
- Mithilfe der Zugriffsrechte für Dateien (s. vorherige Seite) kann man also auch den Zugriff auf die Hardware bestimmten Benutzern vorbehalten. Beispiele: Modem, Floppy.

8.5 Betriebssyteminterna

- Das proc-Verzeichnis enthält imaginäre Dateien mit wichtigen Informationen über die Hardware, das Betriebssystem und die laufenden Programme. Teilweise läßt sich der Betriebssystemkern über das /proc-Verzeichnis konfigurieren.

  Bsp.'e: »cat /proc/cpuinfo« (Hardware)
  »cat /proc/pci« (Betriebssystem)

- Die Datei in '/proc' belegen keinen Platz auf der Festplatte.

8.6 Kopien - einmal anders

- Soll eine Datei oder ein Verzeichnis mehrere Namen erhalten, so legt man statt einer Kopie einen "link" an.

  Bsp.'e: »cd /usr/bin; ls -l mdir mtools mformat mcopy«
  »ls -ld /usr/adm«
  »symlinks -rv /dev«

- Ein "symbolischer link" zeigt auf die die eigentliche Datei (genauer: der link enthält den Namen der eigentlichen Datei). Erst beim Zugriff auf den symbolischen link wird überprüft, ob die eigentliche Datei existiert und ob der Zugriff auf die eigentliche Datei erlaubt ist.
  Bsp.: »ln -s ~/mail ~/Mail; ls -ld ~/Mail«

- Es wird unterschieden zwischen "relativen" und "absoluten" symbolischen links:
  - "absolute links" beziehen sich auf das Wurzel-Verzeichnis
    Bsp.: »ln -s /etc/profile /tmp/profile«

  - "relative links": beziehen sich auf das Verzeichnis, in dem der link angelegt wird
    Bsp.: »ln -s ../etc/profile /tmp/profile«

- Ein "hard-link" ist mit der Orignal-Datei identisch, d.h. nach der Erzeugung des hard-links kann man nicht mehr unterscheiden, welches die Original-Datei war.
  Bsp.: »echo "Hallo Welt" > hallo_welt;\
  ln hallo_welt hello_world; ls -l hall* hell*«

- Daß es sich bei einer Datei um einen hard-link handelt, erkennt man am "link-counter" (siehe 7.5, Abb. 1).
- Löscht man einen hard-link, so wird der link-counter lediglich um 1 heruntergezählt (bei "0" ist die Datei dann tatsächlich gelöscht).

9. Dateibetrachter und Texteditoren

Man will bei einem Texteditor kein WYSIWYG, sondern lediglich die reine Information erfassen, z.B. um eine Konfigurationsdatei zu ändern.

9.1 Lesen von (langen) Textdokumenten

- Die Programme »more«, »less« und »most« dienen zum seitenweisen Anzeigen von Texten, sie sind kleiner und schneller als normale Editoren.
- Nur »less« und »most« lassen sich intuitiv mit den Cursortasten bedienen.
- Mit der Umgebungsvariable 'PAGER' bestimmt man, welches dieser Programme zum Anzeigen von Text benutzt werden soll.

  Bsp.: »export PAGER="less −MMI"«

- Um »more« durch »less« zu ersetzen: »alias more="less −MMI"«

9.2 Programme zum lesen verschiedener Dateiformate

Dateiendungen	Programm	Beispiel
.txt	more	»more German−HOWTO.txt«
.gz	zmore	»zmore German−HOWTO.txt.gz«
		»zcat German−HOWTO.txt.gz | more«
.man .1	man	»man −l German−HOWTO.man«
.dvi	xdvi	»xdvi German−HOWTO.dvi« (X11)
.ps	ghostview	»ghostview German−HOWTO.ps« (X11)

Dekomprimieren von '.gz'−Dateien mit dem Befehl »gunzip«
Bsp.: »gunzip German−HOWTO.dvi.gz«

9.3 Übersicht verbreitete Texteditoren (unvollständig!)

Name	Anfänger	Leistung	Besonderheiten
ae	o++	o+	einfach, winzig, menügeführt
ed	o−−	o	Notfalleditor, einzeilig (!)
vi	o−−	o++++	the king of vi's
emacs	o−	o++++++	nur Kaffekochen kann er nicht
joe	o+	o++	W*rd$tar−kompatibel
mars_e	o+	o++	komplett in Deutsch, nur X11
pico	o+	o	Editor des e−mail−Programms »pine«
le	o++	o++	
we	o+	o++	ähnlich dem TP−Editor

9.4 »vi«−ähnliche Editoren

- »vi« ist der Standard−Editor auf den meisten unix−artigen Systemen, er ist NICHT intiutiv bedienbar.
- Verschiedene Ausgaben des »vi«: »elvis«, »nvi«, »vim«, ... (?).
- Der »vi« unterscheidet zwischen Kommando− und Eingabemodus.
- Tastenbelegung:

im Kommandomodus:

Tasten	Funktion
x	Zeichen löschen
dd	Zeile löschen
:w	Datei sichern
:q	»vi« beenden
i	in Eingabemodus wechseln

im Eingabemodus:

Tasten	Funktion
Esc	Kommandomodus einschalten

9.5 Der Editor »emacs«

- »emacs« ist das Schweizer Taschenmesser unter den Texteditoren; er ist ebenfalls nicht intiutiv bedienbar (dies ist aber nicht wie beim »vi« in einem besonderen Konzept begründet).
- Konfigurationsdateien sind '/usr/lib/emacs/site−lisp/site−start.el' (globale Einstellungen) und '~/.emacs' (persönlich).
- Jede Datei wir in ein eigenes Fenster geladen, das als "Buffer" bezeichnet wird.
- Die vorletzte Zeile des Bildschirms (invertierte Darstellung; Bezeichnung: "mode−line") enthält Informationen über die Datei und den eingestellten Bearbeitungsmodus.
- Die letzte Zeile des Bildschirms wird "minibuffer" genannt und dient der Eingabe von Kommandos oder dem Dialog mit dem Benutzer;
- Wichtige Tastenkombination: <Alt−x> describe−mode <return>
- Deutsche Kurzbeschreibung unter

  http://www.inf.tu−dresden.de/~sr1/projects/emacs−refcard/

Tasten	Funktion	Tasten	Funktion
C−g	Vorgang abbrechen	C−x C−s	Buffer speichern
C−s	Wort suchen	C−x C−c	Beenden/Verlassen
C−_	Undo	C−x C−f	Datei laden
C−w	Text ausschneiden	C−x C−i	Datei einfügen
C−SPC	Marke setzen	C−x 1	Fenster schliessen
C−y	Text einfügen	C−h m	Bearbeitungsmodus anzeigen
F10	Menü (emacs−19.30)		

10. Dokumentation

Es kann nicht oft genug betont werden, daß die vorhandene Dokumentation den größten Teil der "Anfängerfragen" beantworten; meist besteht sie nämlich aus nichts anderem als "Anfängerfragen" (natürlich samt den gesammelten Antworten).
Wer diese Informationsquellen ignoriert, der macht sich selbst (!) und anderen Menschen unnötige Mehrarbeit. In diesem Sinne: RTFM (siehe 46.7)

10.1 Dateinamen und Bezeichnungen, die Hilfe versprechen

- ‘FAQ’: "Frequently Asked Questions" ist eine Sammlung von häufig gestellten Fragen samt Antworten
- ‘HOWTO’: eine überarbeitete FAQ (gleiche Information, Frage/Antwort−Charakter entfällt)
- ‘Guide’: ein deutlich erweitertes HOWTO

- ‘README’: allgemeine Hinweise zu einem Programm
- ‘INSTALL’: Installationsanleitung
- ‘.c’: Programmtext (enthält häufig unmittelbar am Anfang oder am Ende Hinweise)

10.2 Wo findet man Dokumentation?

- Auf der eigenen Festplatte in den Verzeichnissen

  ‘/usr/doc’ (nach Programmen sortiert) und
  ‘/usr/src/linux/Documentation’.

- Im WWW: einen Rundumschlag zum Thema Linux bietet das "Linux Documentation Project" (LDP); erreichbar unter der URL:

  http://www.ldp.org/

- In der newsgroup "comp.os.linux.answers" (siehe 44.6).
- Im Buchhandel.
- Auf "sunsite−mirrors" (siehe 11.3) im Verzeichnis ‘doc’.

- In den FAQ−Archiven z.B. unter
  ftp://www.uni−paderborn.de/FAQ/comp/os/linux

In diesen Archiven findet man nicht nur die Linux−spezifische Dokumentation, sondern auch Informationen über PC−Hardware, PC−Software, andere Betriebssystemen, Programmiersprachen usw.

Die Gliederung erfolgt nach newsgroups (siehe 44.2), d.h. allgemeinere Information findet man z.B. unter
ftp://www.uni−paderborn.de/FAQ/comp/sys/ibm−pc/hardware
ftp://www.uni−paderborn.de/FAQ/comp/unix/questions
ftp://www.uni−paderborn.de/FAQ/comp/windows/x/announce

10.3 "manual−pages":

- "man−pages" sind per »man«−Befehl abrufbare Programmbeschreibungen.
  Bsp.: »man 1 mtools« (siehe 8.2)
  »tkman &« (unter X11)
- Referenzen auf man−pages werden im Format "manpage(sektion)" angegeben.
  Bsp.: »mtools(1)«
- Stichwortsuche in den Titeln der man−pages mit der Option ’−k’:
  Bsp.: »man −k user«
- Am Schluss einer »man−page« findet man oft Beispiele und Tips, sowie die Namen der Konfigurationsdateien.
- Thematische Gliederung der man−pages (Zahlen bezeichnen Sektion):

  1: Benutzerprogramme — 2: Betriebssystemroutinen
  3: C Bibliotheksfunktionen — 4: Geräte
  5: Beschreibung (Format) der Konfigurationsdateien
  6: Spiele — 7: Vermischtes
  8: Systemverwaltung — n: Neues (?)

Insbesondere die Details zu den in diesem Skript mehr beiläufig erwähnten Programmnamen kann man in den man−pages nachlesen.

10.4 Programmparameter, die eine Kurzbeschreibung anzeigen

- ’−?’, ’−h’, ’−−help’, sind die verbreitesten Optionen.
  Bsp.: »ls −−help«
- ’−v’ oder ’−V’ spucken meist die Versionsnummer aus.
  Bsp.: »gcc −v«
- Manche Programme zeigen die Kurzbeschreibung bei Fehlbedienung automatisch an.

10.5 GNU−TeXinfo

- "GNU−TeXinfo" ist eine auf Hypertext basierende Dokumentation.
- Die Bedienung ist nicht intiutiv; die eingebaute Hilfe sollte als erstes gelesen werden ("Alt−x describe−mode <return>").
- Start des Programms mit dem Befehl »info« oder im Editor »emacs« mit <Strg−h> <i>.

10.6 Bücher und Zeitschriften

Eine Liste mit deutschsprachiger Literatur rund um Linux ist erhältlich unter
http://home.pages.de/~joey/Linux/Dynamic/Buecher/

Darüberhinaus erscheinen mir erwähnenswert:

- "Linux−Journal", SSC Inc. (Verlag), engl., monatlich
- "Linux Magazin", dt., monatlich
- "Essential System Administration" (Frisch)
- "Linux", Addison−Wesley (Kofler)
- "Intranetworking mit Linux" (Trümper)

11. Beschaffung von Software

An einigen Stellen erwähne ich Software, ohne genauer zu sagen, woher man diese beziehen kann und wie man diese installiert.
Eine erfolgsversprechende Antwort wäre im Einzelfall etwa so lang wie die folgenden zwei Seiten. Die einzige Möglichkeit zur Lösung des Dillemas besteht also in der Offenlegung des dahinterstehenden, allgemeinen Prinzips.

11.1 Überblick über die verfügbare Software

- "Linux Documentation Project" (LDP) (siehe 10.2)
- Bunte Zusammenstellungen von Hinweisen

  http://www.fokus.gmd.de/linux/linux−prog.html

- Naturwissenschaften, Mathematik, Informatik

  http://chpc06.ch.unito.it/linux/linux_old.html
  http://www.fu−berlin.de/~clausi

- Hochleistungs−Syteme mit Linux

  http://www.cs.cornell.edu/home/mdw/hpc/hpc.html

- "The Linux Midi + Sound Pages"

  http://www.digiserve.com/ar/linux−snd/

- Die "Linux Software Map" (LSM) ist eine allgemeine, aber unvollständige Übersicht mit einheitlichen Beschreibungen den Programmpaketen, man findet sie

  − in Dateiform als "LSM.Datum.gz" auf sunsite−mirrors
  − im WWW als Datenbank, z.B. unter der URL
  http://www.boutell.com/lsm/lsmbykey.cgi
  http://www.iki.fi/liw/linux/cola.html

- Einen Überblick über die erhältlichen kommerziellen Programme gibt das ‘Commercial−HOWTO’.

11.2 Ankündigungen von Neuerungen und Korrekturen

- Ankündigungen von allgemeinem Interesse kann man ca. wöchentlich in der newsgroup "comp.os.linux.announce" nachlesen (siehe 44.1). Das Archiv aller Ankündigungen findet man unter der URL
  http://www.iki.fi/liw/linux/cola.html

- Die e−mail Verteiler der Entwicklergruppen enthalten Hinweise auf Programmteile, die noch nicht für die allgemeine Benutzung geeignet sind.

11.3 Fundorte für Software

- Software wird im Internet auf sogenannten "ftp−Servern" gelagert.
- Folgende ftp−Server haben für die Linux−Gemeinde besondere Bedeutung:

sunsite.unc.edu	die Sammelstelle für Linux−Software schlechthin
tsx−11.mit.edu	ALPHA− und BETA−Software (siehe 13.4)
ftp.kernel.org	neuste Kernelversionen und −patches

- Viele ftp−Server in Deutschland spiegeln diese Softwarearchive (mit nächtlicher Aktualisierung); Bezeichnung: "mirror−sites".
- Außerdem sind die Linux−Bereiche dieser ftp−Server auch auf CD−ROMs gepreßt im (Buch−)Handel erhältlich.
- Nicht alles findet man auf tsx−11 oder sunsite; das ganze Internet ist voll mit Programmen, die nicht speziell für Linux geschrieben sind, die aber einwandfrei unter Linux laufen.

11.4 Zugriff auf ftp−Server

- Solange man die brandaktuellen Versionen nicht wirklich benötigt, sollte man das Internet schonen und CD−ROMs verwenden.

- Der Zugriff auf FTP−Server ist besonders einfach mit den Programmen »mc« ("midnight commander", <F9> "FTP link") oder »netscape«.

- Falls diese nicht zur Verfügung stehen: »ftp« benutzen.

  Aufruf: "ftp FTP−Server"

  Name zum Anmelden: "ftp" oder "anonymous"
  Password: die eigene Email−Adresse

Kommandos:		
	»dir«	Dateinamen auflisten
	»cd«	Verzeichnis wechseln
	"get zB.tgz"	Datei ‘zB.tgz’ herunterladen
	"put zB.tgz"	Datei ‘zB.tgz’ aufladen
	»quit«	ftp−Sitzung beenden

- Internet−Knigge: nach Möglichkeit nur nahe FTP−Server benutzen.

11.5 Navigation und Suche

- Dateien mit Namen ‘ls−lR’ enhalten eine Liste der Namen aller vorrätigen Dateien auf einem ftp−Server.
  Bsp.: »zgrep cdwrite ls−lR.gz«

- Um im Internet komfortabel nach Software zu suchen, eignen sich die Programme »archie« oder »xarchie«; diese basieren auf Inhaltsverzeichnissen vieler großer ftp−Server

  Bsp.: »archie −h archie.th−darmstadt.de mc |more«
  »xarchie &«

12. Installation von Software

12.1 Entwicklungsstadien

- Grundsätzlich ist keine Software fehlerfrei.
- "ALPHA" ist ein Hinweis auf Software, die noch in der Entwicklung steckt und noch nicht ausreichend getestet worden ist; die Wahrscheinlichkeit von schwerwiegenden Programmfehlern leicht erhöht. (Nicht zu verwechseln mit der Computerarchitektur gleichen Namens.)
- "BETA" kennzeichnet Software, die das ALPHA–Stadium (hoffentlich) hinter sich hat und nun einer intensiven Testphase unterzogen wird.
- "stabil" (engl. "stable") wird Software genannt, wenn die BETA–PHASE hinreichend lang war und innerhalb eines gewissen Zeitraums keine neuen, gravierenden Fehler gefunden wurden.

12.2 Programmpakete

- Programmpakete sind i.d.R. mit »tar« gepackt und mit »gzip« komprimiert, und enden mit '.tar.gz' oder kurz '.tgz'; dieser Typ von Dateien kann mit dem Befehl »tar xzvf« ausgepackt werden.

```
        Bsp.:   »tar  tzvf  dosemu0.60.tgz  −C ~/src«
                             ' :: '
        nur Testen −'  ::  '− die folgende Datei (file)
     dekomprimiere −−−'  '−−− sag was du machst (verbose)
```

  (Um tatsächlich auszupacken: statt 't' ein 'x' verwenden.)

- Die weniger weit verbreitete Verpackungsart »shar« ("shell archive") wird mit dem Befehl »unshar« entpackt.

- Programmpakete können sowohl fertig compilierte Programme enthalten ("binaries"), als auch Quelltexte ("source–code"). Letztere müssen zunächst noch mit einem Compiler in die Sprache des Prozessors ("Penguin") übersetzt werden ("compilieren").
- Quelltexte entpackt man am besten in den Verzeichnissen '/usr/local/src' oder '~/src'; evtl. muss man vorher Unterverzeichnisse anlegen.

- Übersicht der Dateien, die typischerweise in Programmpaketen enthalten sind

Name(n) oder Dateiendung(en)	Bedeutung
README	Überblick ("lies mich")
INSTALL	Anleitungen zur Installation
doc	Unterverzeichnis mit Dokumentation
Imakefile	siehe ????
Makefile	siehe 12.4
.c	Programmtexte in der Programmiersprache "C"
.h	Variablendeklarationen, usw.
.1 .man	manual–pages (siehe 10.3)
.ad	Voreinstellungen für X–Resources (siehe 51.)

12.3 Korrekturen und Flickwerkzeug

- Sind (kleinere) Korrekturen an Programmen notwendig, so werden meist nur die notwendigen Änderungen am Quelltext verteilt, nicht das komplette Programm (drastische Reduzierung der Netzbelastung).
- Diese Korrekturen werden als "patches" (dt. "Flicken") oder "diffs" bezeichnet.
- Erzeugung von patches/diffs mit »diff«:

```
        Bsp.:   "diff −u  Datei.alt  Datei.neu > Unterschiede"
```

  Beispiel eines "patches":

```
        −−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
        −−− v1.3.83/linux/Makefile      Wed Apr  3 16:06:54 1996
        +++ linux/Makefile      Thu Apr  4 09:20:50 1996
        @@ −1,4 +1,4 @@
         VERSION = 1
         PATCHLEVEL = 3
        −SUBLEVEL = 83
        +SUBLEVEL = 84
        −−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
```

  Dieser patch besagt: tausche in der Datei 'linux/Makefile' die Zeile "SUBLEVEL = 83" gegen die Zeile "SUBLEVEL = 84" aus.

- "context–diffs": die der Änderung vorangehenden und nachfolgenden Zeilen werden aufgelistet [dadurch erscheinen die Änderungen im Zusammenhang und sind für den Menschen verständlicher].

- Anwenden eines "patches" mit dem Programm »patch«:

```
        Bsp.:   "cat Unterschiede |  patch −p0"
```

12.4 Das Programm »make«

- Der überwiegende Teil der Software für Linux liegt im Quelltext vor und muß noch compiliert werden.
- Das Kommando »make« compiliert/erstellt automatisch alle notwendigen Teile eines Programms. (genauer: es erstellt nur die fehlenden und fügt sie zusammen)
- Wie was zu tun ist erfährt »make« aus der Datei 'Makefile'.
- Liegt dem Programm kein 'Makefile' bei, so gelten die Regeln:

  - Für X11–basierte Programme wird das 'Makefile' automatisch durch den Befehl »xmkmf« erzeugt (setzt ein eine Datei mit Namen 'Imakefile' voraus).
  - Für GNU–Programme kann ein 'Makefile' automatisch durch den Befehl »./configure« erstellt werden.

- Meist reicht das Kommando »make«, um ein Programm so weit zu bauen, daß man es testen kann.
- Durch »make install« wird das Programm i.d.R. installiert (und mit ein wenig Glück kann man es mit »make uninstall« de–installieren).
- Empfohlene Einstellungen im Makefile: "PREFIX=/usr/local" oder "BINDIR=/usr/local/bin" (weitere entsprechend, siehe ????).

13. Lebenszyklus eines Linux–Systems

Alle Programme und Vorgänge werden am Ende des Manuskripts im Detail besprochen.

Glossar:

- o "booten": Start eines Betriebssystems
- o "hochfahren": umgangssprachlicher Ausdruck für booten
- o "Sektor": 512 byte großer Datenblock; kleinste Einheit zur Organisation eines Datenträgers

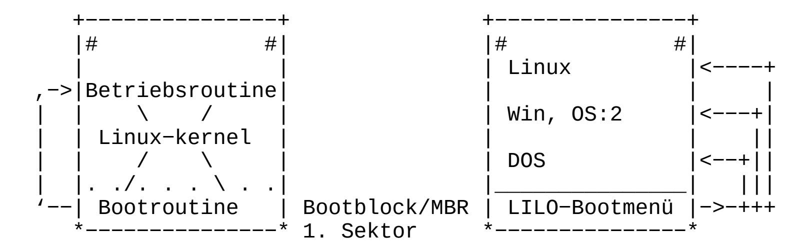
13.1 Der Instinkt der Hardware

- o Nach dem Einschalten des Computers gehen alle Komponenten in einen Ausgangszustand und führen teilweise Selbstests durch.
- o Nach dem Abschluss aller Selbsttests beginnt der Prozessor automatisch mit der Ausführung der BIOS–Routinen:
  - – BIOS = "basic input/output system"
  - – in einem Chip auf der Hauptplatine des PCs dauerhaft gespeichert
  - – Konfiguration durch das "BIOS–Setup"
  - – lädt automatisch die Anweisungen aus dem allerersten Sektor der Diskette oder der Festplatte

13.2 Der erste Sektor eines Datenträgers

- o Hier befinden sich Anweisungen zum Start eines Betriebssystem.
- o Linux kann in diesen Bereich den Boot–Manager »lilo« ("Linux Loader") installieren, der die Wahl zwischen verschiedenen Betriebssystemen erlaubt (siehe ????).
- o Bedienung des Boot–Managers »lilo«:
  - – Taste <Alt> zum Aktivieren des Boot–Prompts
  - – Taste <Tab> listet die Namen der bootbaren Betriebssysteme auf (vom Benutzer konfiguriert)
  - – durch Eingabe von Parametern am Boot–Prompt läßt sich der Bootvorgang des Betriebssystemkerns beeinflussen; siehe BootPrompt–HOWTO

- o Linux kann auch durch die Boot–Manager der anderen Betriebssysteme gebootet werden.



(Die Diskette soll in dieser Abbildung symbolisch für einen Datenträger stehen. Details Seite 53)

13.3 Der Linux–Betriebssystemkern

- o Ziel der vorhergehenden Aktionen ist es, den Betriebssystemkern von Linux in den Arbeitsspeicher zu laden und zu starten.
- o Der Betriebssystemkern soll alle Resourcen des Computers (Rechenleistung, Arbeitsspeicher, Netzwerkkomponenten) zwischen den konkurierenden Programmen (–> multitasking) aufteilen.
- o Dazu muß der Betriebssystemkern diese Resourcen suchen, finden und initialisieren; die meisten Meldungen rühren daher.
  [Mit dem Programm »dmesg« kann man diese Meldungen nach dem Booten nochmal anzeigen lassen.]
- o Nachdem dies erfolgt ist, wird von einer bestimmten Partition ("root–Partition") das Programm »init« ausgeführt.

13.4 Das Ur–Programm »init«

- o »init« ist Vorfahre aller anderen Programme.
  Bsp.: »pstree«

- o Die eigentliche Arbeit überlässt »init« shell–Skripten in '/etc/init.d/', bzw. '/etc/rc.d/'; diese entsprechen einer verzweigten 'autoexec.bat' von DOS.
- o Wichtige Vorgänge sind:
  - – Überprüfung der Festplatte; sind deren Daten stimmig?
  - – Zusammenfassung aller lokaler (physikalisch) Speichermedien zu einem großen Dateibaum
  - – Aktivierung von swap (Auslagerungsdatei bei nicht ausreichendem Arbeitspeicher [RAM])
  - – Laden zusätzlicher Treiber, sogenannter Module
  - – Aktivierung des Netzwerks
  - – Belegung der Tastatur und Einstellung einer Schriftart für den Textbildschirm
  - – Starten wichtiger (ständig aktiver) Systemprogramme

- o Ist das System bereit, so erscheinen auf den Textkonsolen login–Aufforderungen und evtl. wird die graphische Oberfläche gestartet.

13.5 Laufender Betrieb

- o Was jetzt passiert ist Sache des Benutzers.

13.6 Vor dem Abschalten des Computers

- o Spätestens nach 497 Tagen ununterbrochenem Betrieb muß Linux angehalten werden (Überlauf des 32–bittigen jiffies–Zählers).

- o Während des Betriebs werden nicht alle Daten sofort bei ihrer Erzeugung auf der Festplatte gespeichert; man muß dies also vor dem Ausschalten sicherstellen; dies geschieht mit den Befehlen »shutdown –r now« (reboot) oder »shutdown –h now« (halt).
- o Der "Informatikergriff" <Strg–Alt–Entf> kann auch benutzt werden. [De–Aktivierung in '/etc/inittab'].

14. Der "login"−Vorgang

14.1 Grundlegendes

- Vor dem Zugriff auf einen Linux−Rechner muß sich der Benutzer mit einem Namen und einem Geheimwort ausweisen (es gibt aber noch andere Verfahren).
- Dieses Ausweisen nennt man in der Umgangsprache "einloggen"; engl. "logging in" oder kurz "log in" (dt.: "log Dich ein").
- Der umgekehrte Vorgang, das Abmelden (nie vergessen!) wird als "ausloggen" oder "log out" bezeichnet.

14.2 Das login−Prompt

- Das login−Prompt ist die Aufforderung an den Benutzer, die login−id einzugeben.

```
        Bsp.'e: »FeilSurf login: «              (Konsole/telnet)
                »annex username: «              (Terminalserver)
                »Name (localhost:truemper): «   (FTP)
```

- Programme zur "Erzeugung" von login−Prompts

Name	Begrüßungstext	Beschreibung
getty	/etc/issue	Standard für Textkonsolen
mgetty	/etc/issue	Fax, PPP oder login via modem
xdm	/etc/X11/xdm/Xresources	X11−login (siehe 50.5)
telnetd	/etc/issue.net /etc/issue	telnet−login (Tür zum Internet)
ftpd	/etc/ftpd/ftpaccess	

- Die gettys werden in der Datei '/etc/inittab' aktiviert und von »init« verwaltet.
- Ist der Benutzernamen eingegeben, so wird i.d.R. das Programm »login« (oder Verwandte) gestartet.

14.3 Das Programm »login«

- »login« führt alle notwendigen Schritte zum Anmelden des Benutzers durch (Konsole und telnet); von "getty" und Verwandten erhält es den Namen des Benutzers.

- Auswertung der Zugriffkontrollen / −sperren:

Datei	Funktion
'/etc/usertty'	[Wird die Datei überhaupt ausgewertet?]
'/etc/passwd'	Gibt es eine Benutzer mit diesem Namen?
'/etc/nologin'	Warum dürfen sich keine Benutzer einloggen? (Kann zum Sperren des Systems verwendet werden.)
'/etc/securetty'	Darf sich root an diesem Terminal einloggen?

- Autorisierung: Abfrage des passwords durch »login« ("password−prompt").

  '/etc/passwd'   Stimmt das password? (Vergleich der verschlüsselten Versionen)

- Diverses

  '~/.hushlogin' Will der Benutzer ein login ohne viel Tam−Tam?

- Logbücher: vermerken, daß sich der Benutzer eingeloggt hat

'/var/run/utmp'	Logbuch
'/var/log/wtmp'	Logbuch
'/var/log/lastlog'	Logbuch

- Abschliessende Aktionen
  - Übereignung des Terminals an den Benutzer
  - Ausgabe der Datei '/etc/motd' ("Message of the day")
  - Start der in '/etc/passwd' angegebenen shell

15. "passwords"

15.1 Grundlegendes zu passwords

- Unter Linux wird grundsätzlich nur mit den verschlüsselten Versionen der passwords gearbeitet − das original−password kennt nur der Benutzer.
- Das password wird erst nach der Eingabe von »login« verschlüsselt, d.h. bei login−Vorgängen via Internet wird es zunächst im Klartext übertragen.
- Da die Datei ‘/etc/passwd’ von jedermann lesbar ist, sind es auch die verschlüsselten passwords.
- Durch Verschlüsseln von Wörtern und anschließendem Vergleich kann man passwords also ERRATEN; aus diesem Grund gibt es "einfache" und "schwere" passwords.
- "shadow−passwords" ist ein Verfahren zum Aufbewahren der (verschlüsselten!) passwords in einer nicht öffentlich lesbaren Datei (siehe ‘Shadow−Password−HOWTO’).

15.2 Verschlüsselungstechnik

- Die Standard−Verschlüsselungsverfahren benutzen max. 8 Zeichen des eingegebenen passwords, d.h. "Grundlage" und "Grundlagen" werden nicht unterschieden.
- Von jedem Zeichen werden nur die ersten 7 Bit verwendet, d.h. "Mäander" und "Mdander" sind gleich.
- Somit ergeben sich unter Abzug der nicht eintippbaren Zeichen ´^@´ und ´^?´ nur 126^8= 63.527.879.748.485.376 Möglichkeiten.

- Das verschlüsselte password ist stets 13 Zeichen lang und kann sogar in passwd−Dateien anderer Rechner verwendet werden.
- weitere Informationen siehe "crypt(3)" (manual−page)

15.3 password vergessen: was tun?

- root kann das password eines Benutzers mit dem Programm »passwd« ändern.
- Hat root ihr password vergessen, so muß man rebooten und entweder
  - am LILO−Prompt mit "linux single" in den single−user Modus booten (nicht immer ohne password möglich ...)
  - mit den Installationsdisketten oder Rettungsdisketten booten (ohne tatsächlich zu installieren) und sich so Zugang zur Datei ‘/etc/passwd’ zu verschaffen, um das password von root zu löschen (−> kein password).

15.4 Wahl eines "guten" passwords

- Das Programm »npasswd« läßt leicht zu erratende passwords erst gar nicht zu (Alternative zu dem herkömmlichen »passwd«−Programm).

- Ein password muß 2 Bedingungen erfüllen:
  - es muß einfach zu merken sein, damit man es nicht notieren braucht
  - es darf nicht mit Standard−Software wie »crack« zu erraten sein

- Daher scheiden als password aus:
  - alle Dinge, die in Verbindung mit eigenen Person stehen (Adresse, Telefonnummer, Autokennzeichen, Lieblingsgericht, Name des Haustieres, usw.)
  - alle bekannten Wörter und Abkürzungen, die nur leicht oder in gängiger Form abgewandelt werden

- Übrig bleiben Akronyme: bilde einen Satz und nimm als password jeweils das erste Zeichen eines jeden Wortes (oder ähnlich)

  Bsp.: "Ich esse gerne frisch geröstete Pfirsichkerne"
  −−> "IchegfgP" ("Ich", um 8 Buchstaben zu erhalten)

- Allzu offensichtliche Akronyme sollte man aber ebenfalls nicht wählen (Sprichworte, Titel von Musik−Stücken oder Filmen, usw)

15.5 "Knacken" von passwords

- Die gängigsten 1.000.000 Wörter, Romanfiguren, Städtenamen, usw. lassen sich auf einem P90 in weniger als 60 Minuten durchprobieren; deren beliebtesten Abwandlungen innerhalb von 24h.
- »crack« ist ein Programm zum Erraten von passwords, es bedient sich eines (möglichst umfangreichen) Wörterbuchs und einer Reihe von Regeln zur Abwandlung von Wörtern.
- Der Einsatz von »crack« kann strafbar sein und ist daher mit äußerster Vorsicht zu geniessen.

16. Benutzer und Gruppen

16.1 Benutzer

- o Linux unterscheidet die Benutzer des Computers anhand eines max. 8−stelligen Namens, der "login−id" (oder auch "username").
- o Vor dem Zugriff auf ein Linux−System muß sich der Benutzer mit der login−id und einem Geheimwort ("password") ausweisen.
- o Eine Zugangsberechtigung zu einem Linux−Rechner wird "account" (dt. "Konto") genannt.
- o Die Daten eines Benutzers sind mit dessen Namen versehen und können so von den Daten anderer Benuter unterschieden werden.
- o Jedem Benutzer ist ein privater Bereich im Verzeichnisbaum zugewiesen, das sogenannte home−Verzeichnis (engl.: "home−directory")
  Bsp.: »echo $HOME«
- o Arbeit in einem Fenster beenden mit den Befehlen »logout« oder »exit«.

16.2 Programme zur Manipulation der account−Daten

Programm	Beschreibung
»passwd«	password ändern
»chsh«	"shell" ändern ("change shell")
»chfn«	vollen Benutzernamen ändern (»change full name«)

- o Die Auswahlmöglichkeiten für »chsh« werden durch die Datei ‘/etc/shells’ begrenzt.
- o Neben dem vollen Benutzernamen kann mit »chfn« auch noch die Telefon− und Raumnummer des Büros und die private Telefonnummer abgespeichert werden (im "GECOS−Feld").
- o Evtl. wird die Wahl des passwords auf "sichere" Geheimworte begrenzt.

16.3 Die Benutzerdatenbank ‘/etc/passwd’

- o Die Benutzerdatenbank ‘/etc/passwd’ ist von jedermann lesbar.
- o Neben der login−ID bekommt der Benutzer noch zwei (mehr oder weniger willkürliche) Nummern zugewiesen, die "user−ID" und die "group−ID".
- o Für interne Zwecke wird stets die "user−ID" verwendet, nicht die login−ID.
- o Das GECOS−Feld ist für Kommentare gedacht und enthält meist den vollen Benutzernamen.
- o Zentrale Verwaltung der Benutzerdatenbank bei mehreren Rechnern mit "NIS" (vormals "YP") oder "NYS" (Linux−Version von "NIS+").
- o Beispiel eines Eintrags in ‘/etc/passwd’:

```
truemper:LkesGuNvQ:998:970:Winfried Truemper,,,,:/home/truemper:/bin/bash
   |         |      |   |                |         |              |
login−ID     |      |   group−ID    GECOS−Feld     |              shell
       password   user−ID                      home−Verzeichnis
```

16.4 Gruppen

- o Gruppen dienen dem gemeinsamen Zugriff mehrerer Benutzer auf dieselben Dateien (bzw. Geräte).
- o Namen von Gruppen dürfen ebenfalls nur 8 Zeichen lang sein (eine häufige Fehlerquelle).
- o Jeder Benutzer ist automatisch Mitglied der in der Datei ‘/etc/passwd’ angegebenen Gruppe.
- o »chgrp« (change group; Gruppenzugehörigkeit einer Datei ändern; nur ausführbar von root oder Mitgliedern der Gruppe).
- o Zusätzliche Mitgliedschaften werden in der Datei ‘/etc/groups’ eingerichtet; das Format ist:

```
                floppy:*:25:truemper,winni
                   |   |  |     |
     Name der Gruppe   |  |     login−IDs der Mitglieder
                password  group−ID (interne Nummer der Gruppe)
```

16.5 Die Systemverwalterin "root"

- o Die Systemverwalterin root (auch "Superuserin" genannt) darf ALLES, sie hat die user−ID 0 und die group−ID 0 und wird unter der login−id "root" geführt.
- o Insbesondere die Konfiguration des Systems ist root vorbehalten, dazu gehören:
  - − Anlegen von Benutzern und Gruppen
  - − Konfiguration von Hardware
  - − Festlegung der Betriebsparameter
- o Die meisten Programme in den Verzeichnissen ‘/sbin’, ‘/usr/sbin’ und ‘/usr/local/sbin’ sind nur von root ausführbar; aus diesem Grund sind diese Verzeichnisse bei gewöhnlichen Benutzern auch nicht im Suchpfad für ausführbare Programme ("$PATH", siehe 22.4) enthalten.
- o Programme zur Benutzerverwaltung

Name	Beschreibung
»chown«	Eigentümer einer Datei ändern ("change owner") Bsp.: »chown −R mueller /home/mueller«
»adduser«	neuen Benutzer anlegen
»groupadd«	neue Gruppe anlegen
»sudo«	begrenzte Vergabe von root−Rechten an Benutzer

16.6 Benutzerstatistiken und −informationen

- o wer macht was, wo und seit wann: »w«, »who«
- o wer war zuletzt da: »last −n 10«
- o der gläserne Benutzer: »finger«
  Bsp.: »finger −l $LOGNAME«
- o »ac«, »sa«, »sac«, »lastcomm«: Auslastung des Rechners durch einen Benutzer über einen bestimmten Zeitraum (siehe »info accounting«)
- o »pac«: wieviele Seiten hat ein Benutzer gedruckt?

17. Der Bildschirm

17.1 Das Problem

o Jedes Programm und jeder Benutzer braucht eine Möglichkeit zur Ausgabe von Daten und Fehlermeldungen, d.h. multiuser/multitasking erfordern mehr Monitore, als physikalisch vorhanden sind.
o Unter Linux kann man bis zu 4 VGA−Monitore an den PC anschliessen, sowie serielle Terminals, aber selbst das würde nicht ausreichen, wäre zu teuer, unkomfortabel und zu aufwendig.
o Die Lösung: virtuelle Konsolen (und virtuelle Terminals)

17.2 Virtuelle Konsolen

o Die "Konsole" ist die Kombination aus direkt an die Workstation angeschlossener Tastatur und dem Monitor.
o "virtuelle Konsolen" sind den Programmen vorgetäuschte Konsolen, die vom Benutzer wahlweise auf der Konsole eingeblendet werden können.
o Eigenschaften:
  − <Alt−Ctrl−F1> − <Alt−Ctrl−F8> schalten hin und her
  − Zurückblättern mit <Shift−Bild> möglich (aus Sicherheitsgründen nicht mehr nach Wechsel der Konsole)
  − Cut & Paste möglich mit "Maustreiber" »gpm«
  − virtuelle Konsolen sind text− und grafikfähig
  − VESA Powersave−Modus wird unterstützt (siehe Datei ‘/usr/src/linux/drivers/char/vesa_blank.c’)

o zugehörige Gerätedateien sind ‘/dev/tty0’ − ‘/dev/tty8’ (siehe 30.6)
o Linux kann maximal 63 virtuelle Konsolen verwalten, vorkonfiguriert sind aber nur 8.
o Aktivierung der login−Aufforderungen an den virtuellen Konsolen in ‘/etc/inittab’.
o Programme auf "leeren" (unbenutzten) virtuellen Konsolen starten mit dem Programm »open« (siehe "ManyConsoles mini−HOWTO")
  Bsp.: »open −sl −− bash −login«

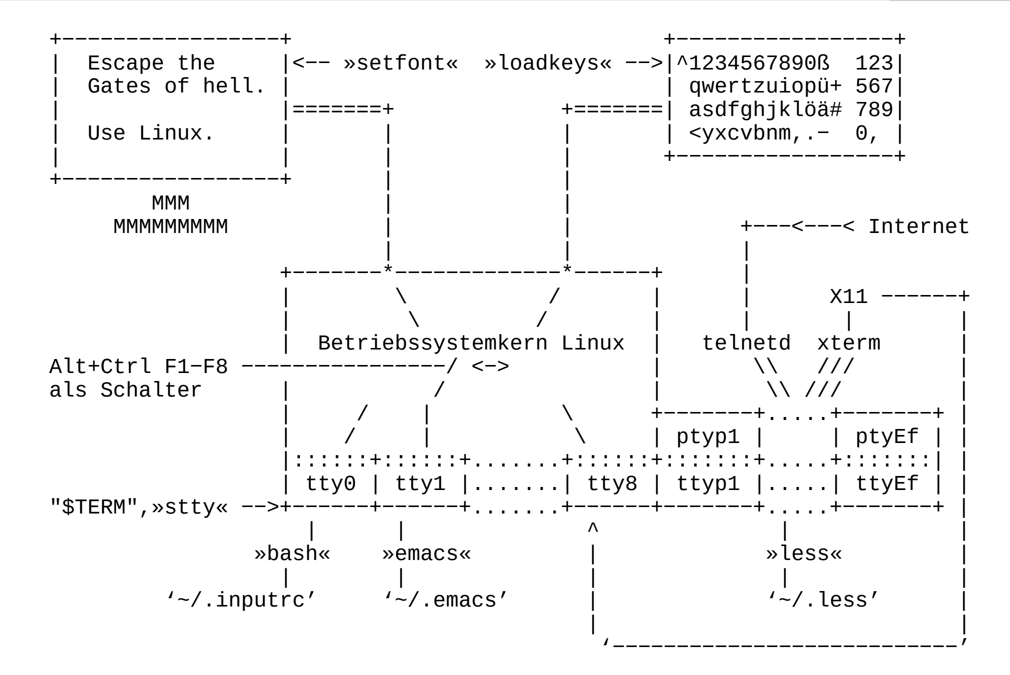
17.3 Konfiguration

o Der Befehl »setfont« erlaubt die Wahl einer anderen Schriftart durch Laden einer Datei aus dem Verzeichnis ‘/usr/lib/kbd/consolefonts’; für Deutschland wählt man.
  »setfont lat1u−16.psf« (‘/etc/rc.boot/console’)

o Höhere Auflösungen im Textmodus durch
  − den Eintrag "vga=ask" in ‘/etc/lilo.conf’ (siehe ????)
  − das Programm »resizecons« (Experimente notwendig, Vorsicht)
  − das Paket "SVGATextMode" (evtl. flimmerfreies Bild)

o Photo eines Textbildschirms ("screen shot") möglich (siehe 30.6)
  Bsp.: »cat /dev/vcs1 > screen_dump«

o Erscheinungsbild der virtuellen Konsolen ändern: »setterm«
  Bsp.: »setterm −foreground green −store −clear«



17.4 Ansteuerung der virtuellen Bildschirme

17.4.1 Der Textmodus

o Im Textmodus ist der direkte Zugriff auf den Bildschirmspeicher nicht erlaubt (siehe 20.1).
o Stattdessen gibt es spezielle Steuerzeichen, die nicht als Zeichen auf dem Bildschirm dargestellt werden, sondern die bestimmte Funktionen auslösen.
  Bsp.: »echo −e "\033[2;20H Hallo Welt "; sleep 5«

o Der Betriebssystemkern setzt die Steuerzeichen in Anweisungen für den Bildschirmspeicher um, er bedient sich dabei des "vt102−Standards". Siehe 20.2

17.4.2 SVGA−Modus

o Linux bietet Routinen zur direkten Nutzung des SVGA−Modus der Grafikkarte. Nur wenige Programme nutzen diese Routinen (»gnuplot«, »gs«, Spiele) − meist aber nur als Option, da die Netzwerkfähigkeit fehlt und die Ausführung auf anderen Architekturen als dem PC nicht möglich wäre.
o Konfiguration des SVGA−Modus unter ‘/etc/libvga.config’.

17.4.3 X−Window−System

o Das X−Window−System ist ein netzwerkfähiges Grafiksystem, das für viele unix−artige Systeme zur Verfügung steht. Siehe 49.1

18. Die Tastatur

18.1 Das Problem

- Wegen der Netzwerkfähigkeiten von Linux wird ein Programm evtl. mit einer Tastatur bedient, die nicht an der eigenen Maschine angeschlossen ist. Eine direkte Abfrage der Tastatur ist also nicht immer möglich und daher nicht angebracht.
- Außerdem ist der direkte Zugriff auf die Hardware zur Vermeidung allgemeinen Chaos i.d.R. dem Betriebssystemkern vorbehalten (man denke insbesondere an die "konkurierenden" virtuellen Konsolen).

18.1.1 Die Tastatur im Textmodus

- Im Textmodus erhält ein Programm jeden Tastendruck als (Folge von) Zeichen geschickt.
  [Und da man Zeichen wiederum sehr gut durch das Internet schicken kann, lassen sich unter Linux auch Programme auf anderen Maschinen im Internet bedienen.]
- Bei gewöhnlichen Buchstaben und Sonderzeichen ist diese Vorgehensweise unproblematisch, aber die Cursortasten und die Tasten zum Löschen von Text müssen als besondere Zeichen(−folgen) an das betreffende Programm übermittelt werden.

- "Escape−Sequenzen": Sobald ein Programm das Zeichen "Escape" (\033) empfängt, wartet es die folgenden Zeichen ab und interpretiert sie als Taste mit Sonderfunktion. Diese Zeichen werden dann nicht auf dem Monitor dargestellt. Bsp.:

Taste	Pos1	Einfg	Entf	Ende	Seite hoch/runter	
Zeichenfolge	ESC[1~	ESC[2~	ESC[3~	ESC[4~	ESC[5~	ESC[6~

- "Control−Zeichen": Das Alphabet des Menschen beginnt erst an der 65. Stelle im Computer−Alphabet. Davor (und dahinter) ist nicht nur Platz für Zahlen und Sonderzeichen, sondern man hat unter Linux sogar die ersten 33 Zeichen als einbuchstabige "Steuerbefehle" reserviert.

Taste	Pos1		Entf	Ende	Backspace	
Zeichenfolge	Strg−A		Strg−D	Strg−E	Strg−H	Strg−?

18.1.2 Tastatur im SVGA−Modus (virtuelle Konsolen)

- Spiele für den SVGA−Modus der virtuellen Konsolen fragen die Tastatur oft direkt ab (Paket »libkbd«).

18.1.3 Das X−Window−System

- In »xterm«−Fenstern gilt das unter 16.1.1 Gesagte (ein »xterm« simuliert nämlich eine virtuelle Konsole).
- Alle "echten" X−Programme bekommen die Eingaben von der Tastatur über symbolische Namen für die Tasten mitgeteilt.

18.2 Konfiguration der Tastatur

- Die Tastatur der Text− und SVGA−Konsolen wird durch den Befehl »loadkeys« konfiguriert.
  Bsp.: »loadkeys de−latin1−nodeadkeys.map«

- Dieser Befehl weist den Tasten KEINE Funktionen zu, sondern nur bestimmte Zeichenfolgen (d.h. man muß diese Zeichenfolgen u.U. noch den Programmen beibringen). Beispiele: siehe 19.1
- Die zugehörigen Konfiguratationsdateien liegen im Verzeichnis ‘/usr/lib/kbd/keytables’ und können mit einem gewöhnlichen Editor bearbeitet werden. Dokumentation mit »man 5 keytables«

- Zur Tastaturbelegung des X−Windows−Systems siehe 51.

18.3 Spezialitäten

Taste	Name	Funktion
<Strg−Rollen>	Show_State	Übersicht der laufenden Prozesse
<Shift−Rollen>	Show_Memory	Statistik zur Speicherauslastung
<AltGr−Rollen>	Show_Registers	Register−dump (das sieht man gerne!)
<AltGr−BS>	SAK	Secure Attention Key
<Strg−Alt−Entf>	Boot	Booten
<Shift−Bild>	Scroll_Backward	Bildschirmseite zurückblättern
<Shift−Bild>	Scroll_Forward	Bildschirmseite vorblättern
<Alt−F1>	Console_1	Umschalten auf Konsole Nr. 1
<Alt−F12>	Console_12	Umschalten auf Konsole Nr. 12
<AltGr−F1>	Console_13	Umschalten auf Konsole Nr. 13
<AltGr−F12>	Console_24	Umschalten auf Konsole Nr. 24

- Der "Secure Attention Key" (<SAK>) weist das Betriebssystem an, sämtliche Programme von einer virtuellen Konsole zu verjagen und eine (garantiert) frische login−Aufforderung zu starten.
  [Wer kann bei öffentlich zugänglichen Computer−Pools schon sagen, ob die login−Aufforderung nicht nur vorgetäuscht ist ...].

18.4 Eternal flame: BackSpace und Delete − eine Lösungsmöglichkeit

	Steuerzeichen	Funktion
traditionell:		
Backspace <−−|	^h	oft: lösche linkes Zeichen
<Entf>, <Del>	^?	lösche linkes Zeichen
alternativ		
Backspace <−−|	^?	lösche linkes Zeichen
<Entf>, <Del>	^d	lösche aktuelles Zeichen

- Probleme mit der traditionellen Belegung: <−−| funktioniert nicht immer, z.B. kann man bei der Eingabe des passwords nur mit <Entf> korrigieren und im "unkonfigurierten" emacs (wichtiger Texteditor) zeigt <−−| die Hilfe an (statt zu löschen). Ächz!
- Jetzt muß man den Programmen lediglich noch mitteilen, daß ´^?´ die Funktion "lösche linkes Zeichen" hat:
  »stty erase ^?« (unter Linux fast ein Standard, nur nötig bei Arbeit auf anderen Maschinen)

19. Konkrete Vorschläge zur Tastaturbelegung

19.1 Ergänzungen/Änderungen für ‘de−latin1−nodeadkeys.map’

```
-----8<----- hier schneiden -----8<-----
# Delete = ^? (Name bezeichnet Zeichen, nicht Funktion)
keycode                 14 = Delete     BackSpace
        alt   keycode   14 = Meta_Delete
        altgr keycode   14 = SAK
# wortweises springen mit den Cursortasten
keycode                 105 = Left
        control keycode 105 = Meta_b
keycode                 106 = Right
        control keycode 106 = Meta_f
# franz. Anführungszeichen mit <AltGr-,> und <AltGr-.>
keycode                 51 = comma semicolon guillemotright
        alt keycode     51 = Meta_comma
     shift alt keycode  51 = Meta_less
keycode                 52 = period colon guillemotleft
        alt keycode     52 = Meta_period
     shift alt keycode  52 = Meta_greater
string Find = "\001"            # Anfang der Zeile (emacs)
string Insert = "\033[2~"       # Einfg (vt102)
string Remove = "\004"          # Lösche Zeichen (emacs)
string Select = "\005"          # Ende der Zeile (emacs)
string Prior = "\033[5~"        # vorherige Seite (vt102)
string Next = "\033[6~"         # nächste Seite (vt102)
-----8<----- hier schneiden -----8<-----
```

- Empfohlener Name: ‘de−hack1.map’ (Kopie von Original verwenden)
- Murphy’s Gesetz: wie man die Escape−Sequenzen und Control−Zeichen auch wählt, irgendein Programm wird diese Einstellung NICHT akzeptieren.
- Erfolgsversprechendes Rezept: emacs−Tastenbelegung gemischt mit dem vt100−Standard. Diese Vorgehensweise ist zwar unsauber, macht aber die wenigste Arbeit (ist also im Zweifelsfalle wieder schnell rückgängig zu machen).

19.2 Nachbildung obiger Tastenbelegung im »xterm«

```
-----8<----- hier schneiden -----8<-----
xterm -fn 10x20 -xrm '\
        XTerm*ttyModes: erase ^?  intr ^c  quit ^\\   \
        XTerm.VT100.Translations:  #override \n\
        <Key>Home:      string(0x01) \n\
        <Key>End:       string(0x05) \n\
        <Key>Delete:    string(0x04) \n\
        <Key>BackSpace: string(0x7f) \n\
        Ctrl<Key>Left:  string(0x1b) string("b")\n\
        Ctrl<Key>Right: string(0x1b) string("f")\n'
-----8<----- hier schneiden -----8<-----
```

- Alle anderen Tasten sollten automatisch wie oben belegt sein. (Ansonsten: siehe ????)
- Die Einstellungen zwischen −xrm ’\ und ’ kann man auch permanent als Xresourcen konfigurieren (siehe 51.).
- "0x" = hexadezimale Notation => 0x1b = Escape; 0x7f = Delete

19.3 Konfiguration der Programme

- Die Interpretation der Escape−Sequenzen und Control−Zeichen ist jedem Programm selbst überlassen.

- ‘~/.inputrc’ konfiguriert "readline"−basierte Programme (z.B. »bash«). Um Umlaute mit solchen Programmen verarbeiten zu können, sind folgende Einstellungen notwendig.

```
-----8<----- hier schneiden -----8<-----
set meta-flag on
set convert-meta off
set output-meta on
-----8<----- hier schneiden -----8<-----
```

- Manche Programme werten $TERM aus und halten sich an die Vorgaben in den Dateien ‘/etc/termcap’ oder ‘/etc/terminfo/’ (siehe ????).
- Die meisten Programme kochen ihr eigenes Süppchen.

- Weitgehend akzeptiert werden die (wenigen) Einstellungen des »stty«−Befehls. Z.B. gehört dazu die Festlegung, welches Control−Zeichen als "Backspace" (linkes Zeichen löschen) interpretiert werden soll. Siehe
  »stty −a ; stty −−help | grep CHAR«

19.4 Beispielkonfiguration für den »emacs«

```
-----8<----- hier schneiden -----8<-----
;; Zuerst die Einstellungen für den Text-Modus
;;
(global-set-key "\033[A"  'previous-line)
(global-set-key "\033[B"  'next-line)
(global-set-key "\033[C"  'forward-char)
(global-set-key "\033[D"  'backward-char)
(global-set-key "\033[H"  'beginning-of-line)
(global-set-key "\033[Y"  'end-of-line)
(global-set-key "\033[1~" 'beginning-of-line)
(global-set-key "\033[2~" 'overwrite-mode)
(global-set-key "\033[3~" 'delete-char)
(global-set-key "\033[4~" 'end-of-line)
(global-set-key "\033[5~" 'scroll-down)
(global-set-key "\033[6~" 'scroll-up)
;;
;; jetzt die Einstellungen für den X11-Modus
;;
(global-set-key [delete]   'delete-char)
(global-set-key [home]     'beginning-of-line)
(global-set-key [end]      'end-of-line)
(global-set-key [prior]    'scroll-down)
(global-set-key [next]     'scroll-up)

(global-set-key [C-right]  'forward-word)
(global-set-key [C-left]   'backward-word)
(global-set-key [C-prior]  'beginning-of-buffer)
(global-set-key [C-next]   'end-of-buffer)
-----8<----- hier schneiden -----8<-----
```

20. Terminals

- o Als "Terminal" bezeichnet man ganz allgemein eine Kombination eines Bildschirms mit einer Tastatur.

- o Zur besseren Ausnutzung der sündhaft teuren Großrechner wurden früher massenhaft grüne [?] Textmonitore mit klobiger Tastatur an jene angeschlossen; diese dienten als Terminals.
- o Da unter Linux die Kommandozeile immer noch Dreh− und Angelpunkt ist, haben Terminals nicht an Bedeutung verloren (allerdings hat sich ihr Erscheinungsbild gewandelt).
- o Heutzutage werden pro Monitor mehrere Terminals simuliert, sowohl im Textmodus (Linux−Konsole oder Programm »screen«), als auch unter der grafischen Oberfläche "X11" (Programme »xterm« oder »rxvt«).
- o Die Linux−Konsole simuliert mehrere vt102−Terminals.

- o Terminals sind in erster Linie textbasiert, allerdings gibt es auch Erweiterungen zur Darstellung von Grafiken (tek−Modus).
- o Insbesondere gibt es X−Terminals, diese sind NICHT textbasiert, sondern verstehen das Grafikprotokoll "X11".

20.1 Terminals unter Linux

- o Wie alle Geräte unter Linux sind auch Terminals nur über spezielle Gerätedateien ansprechbar. (Und es gilt der alte Spruch: die Zugriffsrechte von Dateien wirken auch für die Geräte.)

  Bsp.: »echo "Hallo"«
  »(sleep 2; echo "Hallo" > ‘tty‘)&«

- o Der Befehl »tty« zeigt die gerade benutzte Gerätedatei an.
- o Der Eigentümer einer Terminal−Gerätedatei sollte stets der Benutzer sein und die Zugriffsrechte an dieser Datei sollten niemals das Lesen für die Gruppe oder die Allgemeinheit erlauben (ansonsten kann man abgehört werden).
- o Das Schreibrecht an einer Terminal−Gerätedatei ermöglicht anderen Benutzern, eine Nachricht erscheinen zu lassen.
  Bsp.: »mesg y; write $LOGNAME«

- o Löschen des Schreibrechts mit »mesg n«

- o Nur die virtuellen Konsolen von Linux (vt102−Terminals) stehen direkt mit Hardware in Kontakt (VGA−Monitor, PC−Tastatur).

  Gerätedateien: »ls −l /dev/tty[0−9,a−f]

- o Die restlichen Terminals von Linux sind "Pseudo−Terminals", d.h. statt eines Monitors ist eine weitere Gerätedatei nachgeschaltet, die den Output an ein Programm weiterreicht (siehe 17.).
  Erst dieses Programm kümmert sich dann um die Darstellung auf einem Monitor (Bsp.’e: "X11", »screen«).
  Im Extremfall ist dieser Monitor an einem ganz anderen Computer angeschlossen (Bsp.: »telnet«−Sitzung)

  Gerätedateien: »ls −l /dev/tty[pqrstuvwxyzABCDE][0−9,a−f]«

20.2 Ausgabe von Text

- o Wegen der Netzwerkfähigkeiten von Linux sind Bildschirmspeicher und Monitor evtl. in/an einer ganz anderen Maschine, d.h. können u.U. nicht direkt angesprochen werden.
- o Außerdem ist der direkte Zugriff auf die Hardware zur Vermeidung allgemeinen Chaos i.d.R. dem Betriebssystemkern vorbehalten (man denke insbesondere an die "konkurierenden" virtuellen Konsolen).

- o Stattdessen gibt es spezielle Steuerzeichen, die nicht als Zeichen auf dem Bildschirm dargestellt werden, sondern die bestimmte Funktionen auslösen.
  Bsp.: »echo −e "\033[2;20H Hallo Welt "; sleep 5«

  [WARNUNG: Der vt102−Standard ist nur einer von vielen, es ist also eine denkbar schlechte Idee, obige Steuerzeichen tatsächlich in einem Programm zu verwenden. Siehe 20.8]

- o Anders ausgedrückt: Gewöhnliche Buchstaben und Sonderzeichen können direkt ausgegeben werden, Funktionen wie "setzte den Cursor an die 10. Spalte" müssen vom Programm mit Escape−Sequenzen ausgedrückt werden.
- o Diese Escape−Sequenzen können (auch im Textmodus) problemlos mit den eigentlichen Daten über ein Netzwerk von Computern (z.B. Internet) verschickt werden.

20.3 Eingaben über die Tastatur

- o Siehe 18.1. Zugunsten der Verständlichkeit habe ich auf eine strengere Reihenfolge verzichtet.

20.4 Terminaltypen

- o Es gibt Terminals verschiedener Bauart, sie unterscheiden sich in Leistungsfähigkeit und Escape−Sequenzen (siehe 20.2).
- o Der Typ des Terminals wird den Programmen durch die Umgebungsvariable "TERM" mitgeteilt.

  Bsp.: »echo $TERM«
  »export TERM=vt100« (siehe 22.4)

- o Gängige Werte für "TERM" sind:
  "linux": die Linux−Konsole
  "xterm": Fenster mit Kommandozeile unter X−Windows
  "vt100": Standardwert ("grüner Textmonitor")

- o Die Fähigkeiten des Terminals und die zugehörigen Escape−Sequenzen werden beschrieben durch
  − die Datei ‘/etc/termcap’ ("Terminal Capabilities") und
  − die alphabetisch in Verzeichnissen geordneten Dateien unter ‘/usr/lib/terminfo’.

- o Jedem Programm ist es selbst überlassen, ob und mit welcher Methode es diese Informationen auswertet.

20.5 Konfiguration der Terminaleigenschaften

- »stty«: grundlegende Eigenschaften konfigurieren
  Bsp.: »stty −a«
  »stty rows 10 cols 40« (10 Zeilen, 40 Spalten)
  »stty erase ^? susp ^z start ^q stop ^s werase ^w«
  »stty erase ^h« (falls <−| nicht funktioniert)

- Terminal für Nachrichten sperren: »mesg n« (dieses Kommando ändert die Zugriffsrechte an der zugehörigen Gerätedatei)

20.6 Terminal in Ordnung bringen

- Scheint das Terminal "abgestürtzt" zu sein? Meist hat man versehentlich <Strg−S> gedrückt, wodurch die Ausgabe angehalten wird. Ein <Strg−Q> bringt den Datenstrom wieder in Gang.
- Das Kommando »reset« versetzt ein Terminal in einen "brauchbaren" Zustand (notfalls "blind" eintippen).
- Drücken der Tastenkombination <Strg−L> frischt den Bildschirm auf.
- »clear« löscht den Bildschrim (Vorsicht: nur der sichtbare Ausschnitt wird gelöscht, nicht nach oben herausgerutschte Zeilen).
- Falls die tatsächliche Größe (Zeilen mal Spalten) nicht richtig eingestellt ist: »resize«

20.7 Zubehör

- Virtuelle Terminals: das Programm »screen« stellt die gleichen Funktionen wie die Linux−Konsole zur Verfügung (oder war es umgekehrt?); darüberhinaus kann man es "schlafen" legen oder an ein anderes Terminal "umziehen".
- Mit »ttysnoop« kann root Benutzer ausspionieren (der Gebrauch dieses Programms kann strafbar sein).

20.8 Programmierung

- "ncurses": C−Bibliothek zur Bildschirmsteuerung
- "slang": alternative Bibliothek zur Bildschirmsteuerung

- »tput«: elementare Bildschirmsteuerung aus shell−Skripten heraus
  Bsp.: »tput clear; tput cup 2 20; \
  echo "Hallo Welt"; sleep 5«

- »dialog«: Dialog−Boxen für shell−Skripte (hübsch, aber für komplexere Programmen zu unkomfortabel in der Bedienung UND zu umständlich in der Programmierung)
  Bsp.: »dialog −−yesno "Ja oder Nein?" 7 30«

21. ascii−Kunst und −Unterhaltung

21.1 ascii

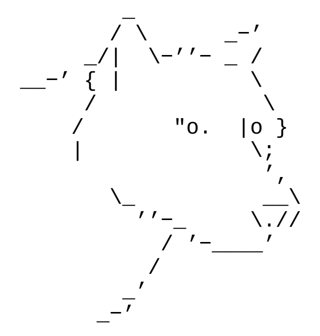
- "ascii" ist ein Standard, der die Buchstaben des englischen Alphabets und einige Sonderzeichen mit Nummern versieht.
- In der Umgangssprache wird "ascii" als Synonym für "nur aus Text bestehend" verwendet (keine Umlaute, usw.!).
- "ascii" umfasst 128 = 2*2*2*2*2*2*2 = 2^7 verschiedene Zeichen; man spricht daher auch von einem 7−bit Zeichensatz.
- Nicht darstellbare ascii−Zeichen: Nr. 0 − 31, 127. Es handelt sich dabei um die control−sequenzen:

```
^@ ^A ^B ^C ^D ^E ^F ^G ^H ^I ^J ^K ^L ^M ^M ^N ^O ^P ^Q ^R
^S ^T ^U ^V ^W ^X ^Y ^Z ^[ ^\ ^] ^^ ^_ ^?
```

- Darstellbare ascii−Zeichen: Nr. 33 − 126

Bereich	Zeichen
32−47	! " # $ % & ’ ( ) * + , − . /
48−57	0 1 2 3 4 5 6 7 8 9
58−64	: ; < = > ? @
65−90	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
91−96	[\ ] ^ _ ‘
97−122	a b c d e f g h i j k l m n o p q r s t u v w x y z
123−126	{ | } ~

21.2 ascii−art



- "ascii−art" ist die Kunst, mit den darstellbaren ascii−Zeichen Bilder zu schaffen
- Beispiele und umfangreiche Dokumentation unter der URL:

  http://gagme.wwa.com/~boba/faq.html>

- Es gibt zahlreiche Werkzeuge zur Erstellung von "ascii−Art", z.B. »figlet«
  Bsp.: »echo "Halo" | figlet«

- "ascii−animations" sind kleine ascii−Filme (Vorsicht, vereinzelt anstössige Inhalte!)

21.3 ascii−Spiele

- Alle textbasierten Spiele haben eine lange Tradition: sie liessen sich schon mit den technischen Mitteln der 80er spielen ...
  Bsp.: »emacs −f gomoku«

- Diese Spiele beziehen Ihren Reiz vornehmlich aus dem Mehrspieler−Modus und der damit verbundenen Kommunikation; meist sind per Internet mehrere dutzend Spieler verbunden.
- Mehr über Spiele im Abschnitt 46.5

22. Die »bash«

22.1 Allgemeines zu shells

- Es gibt unterschiedliche shells (z.B. »tcsh«, »pdksh«, »zsh«), unter Linux ist die »bash« ("bourne again shell") weit verbreitet.
- Die folgende Beschreibung bezieht sich nur auf die »bash«.

22.2 Konfiguration

- Konfigurationdateien für die interaktive Benutzung werden wie shell−Skripte (siehe 24.) geschrieben:

```
        /etc/profile                      (globale Konfiguration)
        ~/.bash_login, ~/.profile         (pers. Konfiguration)
        ~/.bash_logout                    (vor dem Verlassen)
```

```
-----8<----- hier schneiden -----8<-----
EDITOR='emacs'          # siehe 22.4
PAGER='less -MMI'       # siehe 9.1
LESS="-MMI"
LESSCHARSET=latin1
CDPATH=".:..:$HOME"     #
ignoreeof=1             # Zweinmal <Strg-D> drücken zum Ausloggen
export  ignoreeof CDPATH EDITOR LESS LANG LESSCHARSET PAGER

umask u=rwx,g=rx,o=     # Zugriffsrechte für neu erzeugte Dateien
                        # (Widerspruch: entspricht »umask 027« !)
alias ll='ls -l'                        # siehe 22.10
function which () { type -path "$@"; }  # siehe 22.11
export -f which
-----8<----- hier schneiden -----8<-----
```

- Konfigurationsdatei, die nur auf shell−Skripte wirkt:

```
        ~/.bashrc                         (pers. Konfiguration)
```

- Tastenbelegung (siehe ????) und Anzeige von Umlauten:

```
        ~/.inputrc                        (keine Kommentare!)
```

22.3 Eingabehilfen

- Kommandozeileneditor: die eingegebenen Kommandos lassen sich ähnlich wie in einem Editor bearbeiten; siehe Anleitung zum Editor emacs (siehe 9.5).
- Vergangenheitsbewältigung: die shell merkt sich alle ausgeführten Befehle ("command line history"):
  - per Cursortasten kann man herumstöbern
  - gezieltes Suchen mit <Strg−R>
- Tippfaulheit auch ohne Maus möglich:
  - einmal <Tab> vervollständigt Programm− und Dateinamen (engl: "completion"), falls eindeutig möglich
  - zweimal <Tab> zeigt alle Möglichkeiten an

22.4 Umgebungsvariablen

- Umgebungsvariablen beeinflussen sowohl die Kommandozeile selbst, als auch die in ihr gestarteten Programme.
- Setzen von Variablen innerhalb der shell durch einfache Zuweisung.
  Bsp.: »PS1='\h \u \w Kommandozeile:'«

- Setzen mit Vererbung an Kindprogramme durch »export«
  Bsp.: »export TERM=vt100«

- Anzeige aller gesetzten Variablen mit »set« oder »env«
- Löschen mit »unset«
  Bsp.: »unset PATH«

- Beispiele für Umgebungsvariablen

Variablenname	enthält Wert für
CDPATH	Verzeichnisse, die von »cd« durchsucht werden
EDITOR	Bevorzugter Editor
PAGER	Bevorzugter Dateibetrachter
PATH	Verzeichnisse, die nach ausführbaren Programmen durchsucht werden
PRINTER	Bevorzugter Drucker (Standard−Drucker)
PS1	Prompt der bash
PS2	dito auf der 2. Ebene
PWD	"Working Directory" (aktuelles Verzeichnis)
TERM	Terminaltyp
VISUAL	Bevorzugter Dateibetrachter

- Wichtigste Variable: $PATH. Alle in dieser Variablen aufgelisteten Verzeichnisse werden automatisch nach Programmen durchsucht, d.h. anstelle des Kommandos »/bin/ls« ist auch die Kurzform »ls« erlaubt.
  Bsp.: »which ls« (bezieht sich auf $PATH, vgl. 20.1)
  »hash«

22.5 Umleitungen

- Jedes Programm öffnet 3 "Kanäle" (Dateien, siehe 30.13) für die folgenden Zwecke
  Eingabe (0), Ausgabe (1), Fehlermeldungen (2)

- Ohne weitere Vorkehrungen sind diese automatisch mit dem gerade verwendeten Terminal verbunden.
- Eingabeumleitung (aus Datei in Programm) mit ´<´
  Bsp.: »mail truemper < viele_Gruesse«

- Ausgabeumleitung (von Programm in Datei) mit ´>´
  Bsp.: »cat /proc/cpuinfo > meinSystem«
  »cat /proc/meminfo >> meinSystem« (Anhängen)

- Fehlerumleitung mit ´2>´
  Bsp.: »make zImgae 2> Fehler«
  »make zImage &> Fehler« (inkl. Ausgabeumleitung)

22.5 Röhren zur Datenweitergabe

- o Bei einer Röhre (engl. "pipe") erhält das nachfolgende Programm die Ausgabe des vorhergehenden Programms als Eingabe. Eine Röhre ist also ein Umleitung in ein anderes Programm (anstatt in einer Datei).

```
Bsp.:   »cat viele_Gruesse | mail truemper«
        »mail truemper < viele_Gruesse«   (ohne Röhre)
```

- o Röhren können aus mehreren Stücken bestehen.

```
Bsp.:   »grep  bash  /etc/passwd | sort | more«
```

- o Rückgabewert: der des letzten Befehls in der Röhre

22.7 Rückgabewerte

- o Jedes Programm gibt nach Beendigung eine Zahl an die bash zurück, die Auskunft über aufgetretene Fehler gibt.
- o Die Umgebungsvariable ´$?´ enthält diesen Wert − bis zum Ausführen des nächsten Befehls (!).
- o Der Rückgabewert ´0´ bedeutet "kein Fehler".

22.10 Aliase

- o Aliasse dienen zur Umbenennung von Kommandos oder zur Abkürzung von Kommandos.
- o Aliase sind nur im interaktiven Modus wirksam (d.h. nicht in shell−Skripten) und können keine Argumente verarbeiten.

```
−−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
alias ll=’ls −l’
alias la=’ls −la’
alias dir=’ls −l’
alias del=’rm −i’       # bei DOS−Entzugserscheinungen
alias more=’less −MM’
alias unix2dos=’recode lat1:ibmpc’
alias dos2unix=’recode ibmpc:lat1’
alias unix2mac=’recode lat1:applemac’
alias mac2unix=’recode applemac:lat1’
alias talk=’ytalk’
alias weg=’clear; logout’
−−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
```

22.11 Funktionen

- o Funktionen dienen wie aliasse zur Definition "neuer" Kommandos, sind aber auch in shell−Skripten verfügbar und können mit Argumenten aufgerufen werden.

```
−−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
function  which    () { type −path "$@"; }
function  setenv   () { export $1="$2";  }
function  unsetenv () { unset "$@";      }
export −f which setenv unsetenv
−−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
```

22.8 Kommandolisten

- o Man kann mehrere Kommandos hintereinander angeben (verketten), nur muß man der shell mit speziellen Zeichen anzeigen, wo ein Kommando endet und wo das nächste beginnt.

```
Bsp.’e: »make dep && make clean && make zImage« (siehe 55.4)
```

```
Zeichen Bedingung zur Ausführung       Fehler−Rückgabewert
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
&       keine                          stets 0
;       vorheriger Befehl beendet      des zuletzt ausgef. Befehls
&&      vorheriger Befehl erfolgreich  −−−−−−−−−−−− " −−−−−−−−−−−−
||       −− " −−   Befehl mißlungen    −−−−−−−−−−−− " −−−−−−−−−−−−
```

- o Braucht ein Programm eine Möglichkeit zur Eingabe, so muß bei Verwendung von ´&´ diese notfalls per Umleitung aus einer Datei gegeben sein (auch für die Ausgabe empfohlen).

```
Bsp.:   »make zImage &> MakeMeldungen &«
```

23. Ausführen von Kommandos und Ersetzungen

- o Vor dem Ausführen einer Zeile mit Kommandos führt die shell Erweiterungen und Ersetzungen im eingegebenen Text durch. Diese lassen sich z.B. mit »set −x« anzeigen.

	erweitert/ersetzt durch	Beispiel
a{d,c,b}e	alle Kombinationen	»echo a{d,c,b}e«
~	home−Verzeichnis	»echo ~«
		»cd ~root«
$var	Wert der Variablen "var"	»echo $PATH«
$(bfhl)	Ausgabe des Befehls "bfhl"	»rm $(find /tmp −atime +3)«
‘bfhl‘	wie oben	»rm ‘find /tmp −atime +3‘«
$((x+y))	Wert der Operation x+y	»echo $((1+1))«
*	mögliche Dateinamen	»ls −l /dev/lp*«
?	einzelne Zeichen von Dateinamen	
[abc]	Zeichen von Dateinamen	»ls −l /dev/lp[12]«

- o Nach dem Anwenden obiger Regeln werden alle selbst nicht gequoteten Vorkommen der Zeichen ´\´, ´‘´, and ´"´ entfernt.
- o Die sicherste Methode zur Weiterverwendung der Ausgabe eines Befehls als Argumente für einen weiteren Befehl ist das Programm »xargs«; Nachteil: langsamer

```
Bsp.:   »find /tmp −atime +3 −print0 |  xargs −r0 rm −−«
```

[Letzteres ist NICHT möglich mit dem Kommando

```
        »find /tmp −atime +3         |  rm −− «,
```

da die Ausgabe von »find« (eine Liste mit Dateien) nicht als Eingabe an »rm« übergeben werden soll, sondern als Argument]

24. Shell–Skripte

- o Alle genannten Kommandos können auch in eine Datei geschrieben werden, die dann Zeile für Zeile von der shell abgearbeitet wird.
- o Beginnt eine solche Datei mit "#!/bin/sh" und setzt man die Ausführungsrechte (siehe 7.5), so wird die Datei zum Programm.
- o Zu lange Kommandozeilen können mit ´\´ umgebrochen werden (siehe 6.6).
- o Anwendungsbereich: kleinere Aufgaben der Systemverwaltung.
- o Alternativen: andere Skriptsprachen (»python«, »perl«) (Skripte zum Booten müssen mithilfe der bash realisiert werden!).

```
-----8<----- hier schneiden -----8<-----
#!/bin/sh

# Der Eintrag ´#!´ besagt: um den Inhalt dieser Datei auszuführen,
# muß das Programm »/bin/sh« verwendet werden.

# Kommentare beginnen mit dem Zeichen ´#´. Diese Zeilen sind also
# Kommentarzeilen und werden nicht ausgeführt.

# Ein Protokoll aller Fehlermeldungen in die Datei ‘/tmp/tracefile’
# schreiben (siehe ‘/etc/X11/Xsession’)
exec 2> /tmp/tracefile

# Debug–Modus einschalten (Meldungen zählen als Fehlermeldungen)
set –x


# Abschnitt 1: Variablenzuweisung
SERVICE="Winfried"              # einfacher Text
CLIENT=$LOGNAME                 # Wert einer anderen Variablen
GREETING="Hello $CLIENT"        # beides gemischt

HOST=‘hostname‘                 # Ausgabe des Befehls »hostname«


# Abschnitt 2: Ausgabe von Text
echo "$GREETING"                # eine Zeile Text
echo –e "\007\007\007"          # Steuerzeichen mit –e (oktales
                                # Format:
                                #   7 –> \007, 8 –> \010,  9 = \011

cat << EOM                      # alles bis zum EOM (End Of Menu)
        Chez $HOST

        Plez enter a number, honey
No.
1       Margaritha
2       Funghi
3       Mista
4       additional roaches

EOM

# Abschnitt 3: Eingaben durch den Benutzer
read EINGABE                    # Eingabe mit <enter> abschliessen

echo "What else do you want?"
cat – > /tmp/misc               # Eingabe mit <Strg–D> abschliessen
                                # – = Standard–Eingabekanal

# Abschnitt 4: bedingte Ausführung
if [ "$CLIENT" = "gates" ]      # »[« ist ein link auf das Programm
then                            # »test«
    echo "Who let you in?"
fi

case $EINGABE in
    1)
        echo "Pizza Margeritha" >> /tmp/order–$$;;
    2|4)
        echo "The roaches have eaten the mushrooms";;
    3)
        echo "Pizza Mista" >> /tmp/order–$$;;
    *)
        echo "We don’t offer that.";;
esac


# Abschnitt 4: abgezählte Wiederholungen
for  i  in  0 1 2 3 4 5 6 7 8 9 a b c d e f
do
  echo "Wert der Schleifenvariable i ist:  $i"  # einfallsreich!
done

# abgezählt ... naja irgendwie schon!
for i in /dev/lp*
do
   tunelp  $i           # An welcher Gerätedatei hängt der Drucker?
done

# für jede Zeile in einer Datei etwas ausführen
IFS="\
"                               # IFS enthält jetzt das Zeichen
                                # "neue Zeile"
for i in ‘cat /tmp/misc‘
do
  echo "$i is out."
done
IFS=""


# Abschnitt 5: Verarbeitung von Parametern
echo "Der Programname ist: $0"
echo "Die ersten 3 Parameter lauten:  $1  $2  $3"
echo "Sie riefen das Programm mit folgenden Parametern auf: $@"

-----8<----- hier schneiden -----8<-----
```

25. shell−Skripte aus der Praxis

o Dateien finden und löschen (Vorsicht − ein Fehler und das ganze System kann hops−gehen!)

```
-----8<----- hier schneiden -----8<-----
# /etc/cron.daily/standard: standard daily maintenance script

        # Wenn
        #       ‘/tmp’ ein Verzeichnis (’−d’) ist und (’−a’)
        #       ‘/tmp’ kein (’!’) symbolischer link (’−L’) ist und
        #       das Wechseln nach ‘/tmp’ erfolgreich ist

if [ −d /tmp −a ! −L /tmp ] && cd /tmp
then
        # Finde alle Dateien (’−type f’), die länger als 3 Tage
        # nicht mehr angefaßt wurden (’−atime +3’) und reiche die
        # Dateinamen über »xargs« an »rm« (löscht Dateien) weiter.

  find . −type f −atime +3 −print0 | xargs −r0 rm −f −−

        # Finde alle Verzeichnisse (’−type d’), die seit mehr als
        # 1 Tag nicht mehr verändert wurden (’−mtime +3’) und
        # reiche die Dateinamen über »xargs« an# »rmdir« weiter.

  find . ! −name . −type d −mtime +1 −print0 |\
    xargs −r0 rmdir −− >/dev/null 2>&1
fi
-----8<----- hier schneiden -----8<-----
```

o Sicherheitskopien von wichtigen Dateien anfertigen

```
-----8<----- hier schneiden -----8<-----
# /etc/cron.daily/standard: standard daily maintenance script

bak=/var/backups

umask 022               # Geschmackssache!

        # Wenn das Wechseln nach /var/backups erfolgreich ist
if cd $bak
then
        # Vergleiche Sicherheitskopie mit Original und erzeuge ggf.
        # eine aktuelle Kopie

  cmp −s passwd.bak /etc/passwd || cp −p /etc/passwd passwd.bak
  cmp −s group.bak  /etc/group  || cp −p /etc/group  group.bak
fi
-----8<----- hier schneiden -----8<-----
```

o Starten eines daemons

```
-----8<----- hier schneiden -----8<-----
#!/bin/sh
# /etc/init.d/xdm: start or stop XDM.

test −x /usr/bin/X11/xdm || exit 0

case "$1" in
    start)
        start−stop−daemon −−start −−verbose −−exec /usr/bin/X11/xdm
        ;;
    stop)
        start−stop−daemon −−stop −−verbose −−exec /usr/bin/X11/xdm
        ;;
    *)
        echo "Usage: /etc/init.d/xdm {start|stop}"
        exit 1
esac

exit 0
-----8<----- hier schneiden -----8<-----
```

26. Prozesse

26.1 Programme

- Um den Inhalt einer Datei (Skript−Text oder Maschinencode) ausführen zu können, muß das Zugriffsrecht ´x´ ("executable", siehe 7.5) gesetzt sein.
  Bsp.: »chmod a+x Mein_Programm« (z.B. Skript aus 24.)

- Mithilfe der Zugriffsrechte kann die Ausführung eines Programms bestimmten Benutzern oder Benutzergruppen vorbehalten werden.
- Ein Programm wird nach dem Start als "Prozeß" bezeichnet (gebräuchlich ist auch engl. "task").

26.2 Prozesse

- Unter Linux können mehrere (bis zu 512) Prozesse quasi gleichzeitig ausgeführt werden [tatsächlich kann eine CPU natürlich nur ein einziges Programm abarbeiten; durch ein schnelles Abwechseln bei der Programmausführung entsteht die Illusion der Gleichzeitigkeit (Vergleich: Fernsehen)].
- Jedes Programm erhält beim Start eine eindeutige Nummer im Bereich von 4−32767, die sogenannte "process id", kurz "PID".
- Mögliche Zustände für Prozesse sind z.B. R ("running"), S ("sleeping"), Z ("zombie"), etc.
- "zombies" sind bereits beendete Prozesse, deren Daten aber noch für andere Prozesse zur Verfügung stehen müssen.

26.3 Hintergrundprozesse und Daemonen

- Ein "Hintergrundprozeß" ist mit keinem Terminal direkt verbunden.
- Der Befehl »nohup« leitet die Ausgabe eines Programms in eine Datei (‘nohup.out’) um und läßt es mit Priorität 5 starten.
  Bsp.: »nohup make zImage &« (siehe 56.4)

- Ständig laufende Programme werden als "Daemonen" bezeichnet; diese schalten sich von selbst in den Hintergrund und benötigen keine Eingaben vom Benutzer (und verursachen auch keine Ausgabe).
- Fehlermeldungen der Daemonen werden im Verzeichnis ‘/var/log/’ protokolliert (abhängig von ‘/etc/syslog.conf’).
- Die Programmendung ´d´ weisst auf "daemon" hin.

26.4 Verwandtschaftsverhältnisse

- »init« ist der Vorfahre aller Prozesse.
- Startet ein Prozeß einen weiteren Prozeß, so bezeichnet man den ersten als Eltern− und den zweiten als Kindprozeß, den Vorgang des selbst als "forken".
- "fork−bomben" sind Prozesse, die in einer Endlosschleife sich selbst neu starten, sie legen das System u.U. sofort still; dagegen werden unter ‘/usr/src/linux/include/linux/tasks.h’ Vorkehrungen getroffen.

Prozeßbaum (gekürzte Ausgabe von »pstree«)

```
init−+−lpd                Druckmanager ("line printer daemon")
     |−gpm                Maustreiber (Textkonsolen)
     |−3*[getty]          login−Aufforderungen (Textkonsolen)
     |−cron               startet Programme zu einstellbaren Zeiten
     |−xinetd             internet−super−daemon (erweiterte Version)
     |−kflushd            regelmäßiges Leeren von Datenpuffern
     |−klogd              protokolliert Meldungen des kernels
     |−kswapd             Steuerung des Auslagerns von RAM
     |−rpc.portmap        ermöglicht netzwerkweite Prozeduren
     |−syslogd            protokolliert Meldungen versch. Programme
     |−update
     |−bash−−−pine        e−mail−Programm auf der Kommandozeile
     ‘−bash−−−emacs       Texteditor »emacs« auf Kommandozeile
```

26.5 Binärformate

- Ein "Binärformat" ist die Art und Weise der Organisation des Maschinencodes in einer Programmdatei.

Format	Beschreibung
a.out	altes Binärformat von Linux
ELF	Das "Executable and Linking Format" bietet Programmierern wenige, aber enorme Vorteile; die Benutzer profitieren nur indirekt davon (anders ausgedrückt: bemerken es nicht); siehe ‘ELF−HOWTO’.
iBCS2	"Intel Binary Compatibility Standard" ist zur Ausführung von Programmen einiger kommerzieller Un*xe notwendig.

26.6 Speicherverwaltung

- Linux lädt in 4kb−Blöcken nur die Programmteile in das RAM, die tatsächlich benutzt werden (ganz schön geschickt, was?).
- Reicht das RAM nicht zur Ausführung eines oder mehrerer Prozesse aus, so lagert Linux längere Zeit nicht benutzte Programmteile oder Daten auf speziell dafür vorgesehenen Bereichen ("swap−devices") auf der Festplatte aus (Fachausdruck: "paging", auch: "swapping").
- Prozesse, die unerlaubt auf Speicherbereiche (anderer Prozesse) zugreifen wollen, werden SOFORT mit der Fehlermeldung "SEGMENTATION FAULT" ("SIGSEGV", "segmentation violation") abgebrochen.

26.7 Nach einem Programmabsturz ...

- Programmabstürze verursachen i.d.R. einen letzten Schnappschuß des Programms, der in Form der Datei ‘core’ in das aktuelle Verzeichnis geschrieben wird.
- Mit dem Kommando »ulimit« kann man die Erzeugung von core−Dateien unterbinden (bzw. begrenzen).
- core−Dateien können u.U. wiederbelebt werden (als Programme weiterlaufen) oder zur Fehleranalyse mit einem debugger untersucht werden.

27. Prozeßkontrolle

27.1 Informationen über Prozesse

- Im Verzeichnis '/proc' werden Informationen über die Prozesse zur Verfügung gestellt; die Programme »procinfo«, »ps«/»tkps«, »pstree«, »top« und »fuser« werten diese Informationen aus.
- Übergibt man einem Programm ein Argument, so ist mit »ps« und Verwandten sichtbar; deshalb verwendet man passwords nicht als Argument für Programme.
- Der Daemon »rpc.rstatd« stellt Statistiken über die Auslastung der Maschine im Netz zur Verfügung.
- In der Datei '/etc/inetd.conf' sind evtl. weitere, unerwünschte Informationsdienste konfiguriert (»netstat«, »systat«, »finger«).

- »time« kann zum Starten von Programmen verwendet werden, deren Resourcenverbrauch pro Programmlauf man ermitteln will.
  Bsp.: »time −v wc /proc/ksyms«

27.2 Detailierte Ablaufverfolgung

- Höchste Detailstufe erreicht man nur mit einem debugger; dessen Bedienung geht über den Anspruch dieses Manuskripts hinaus.
- »strace« kann zum Starten von Programmen verwendet werden, deren Systemaufrufe protokolliert werden sollen.
  Bsp.: »strace −f −eopen,stat cat /dev/null«

27.3 Signale

- Das Programm »kill« beeinflußt laufende Programme durch Signale.
- Liste aller Signale mit dem Befehl »kill −l«; Auswahl:

Nr.	Name	Taste	Funktion
1	HUP		hangup (Modem hat aufgelegt)
2	INT	<Strg−c>	interrupt (unterbrechen)
3	QUIT	<Strg−\>	quit (verlassen)
9	KILL	<Strg−u>	kill (gewaltsam beenden)
15	TERM		terminate (beenden)
18	CONT		continue (Programm fortsetzen)
19	STOP	<Strg−z>	stop (Programm anhalten)

- Zuordnung von Tasten zu Signalen mit dem Befehl »stty«
- »kill −KILL« kann weder abgefangen, noch ignoriert werden.
- »kill −TERM« ist der Standard und läßt die Chance zum Beenden wichtiger Vorgänge (Speichern, usw.).
  Bsp.: »kill $PPID«

- Die Bedeutung der Signale kann von Programm zu Programm etwas unterschiedlich sein.
- Signale senden unter Angabe des Programmnamens mit »killall«
  Bsp.: »killall −STOP emacs«
  »kill −STOP %emacs"«

- Beenden durch Anklicken: »xkill«

27.4 Seid nett zueinander

- Prozesse können unterschiedlich nett zu anderen Prozessen sein, wenn es um die Zuteilung von Prozessor−Leistung (Rechenzeit) geht.
- Statt Nettigkeit wird auch die Bezeichnung "Priorität" verwendet (umgekehrte Bedeutung!).
- Die Nettigkeit wird durch Zahlen von −20 (keine Rücksicht auf andere) bis 20 (vollkommene Bescheidenheit) angegeben, dazwischen lassen sich alle (ganzzahligen) Abstufungen wählen.
- Die gewöhnliche Nettigkeit liegt bei 0.
- Nettigkeiten <0 sind wegen ihrer Gefährlichkeit nur von root einstellbar; Vorsicht!
- Starten eines Programms mit Nettigkeit ungleich ´0´:
  Bsp.: »nice −n 10 xdos«

- Nachträgliches Verändern der Nettigkeit: »renice«
- Das nachträgliche Herabsetzen der Nettigkeit ist nur der Superuserin root erlaubt.

27.5 Beschränkung der Resourcen

- RAM, Festplattenplatz und Prozessorleistung können durch Fehlfunktion von Prozessen in unbegrenztem Umfang angefordert werden und dadurch kann ein Linux−PC unbenutzbar werden.
- Eine extrem großzügige Beschränkung der Resourcen ist angebracht.
- "disk quotas": Beschränkung des Festplattenplatzes pro Benutzer

27.6 Programme mit speziellen Rechten

- Um Benutzern einen kontrollierten Zugriff auf Daten zu ermöglichen, kann man unter Linux Programme mit speziellen Ausführungsrechten versehen.
- "suid": set−user−id−on−execution, statt ´x´ (siehe 7.5 und 21.1) wird ein ´s´ für den Eigentümer des Programms gesetzt; daraufhin wird das Programm stets im Namen seines Eigentümers und mit seinen Rechten ausgeführt, EGAL welcher Benutzer es startet (vorausgesetzt, er darf es überhaupt starten)

  Bsp.: »ls −l /usr/bin/passwd /etc/passwd«

  obwohl '/etc/passwd' für die Benutzer nur lesbar ist, kann jeder Benutzer seine Daten in dieser Datei mit dem Kommando »passwd« ("suid−root") ändern

- "sgid": set−group−id−on−execution, statt ´x´
  [entsprechende Funktion wie bei »suid«−Progammen]

  VORSICHT: diese Arten von Programmen sind potentielle Sicherheitslücken !
  Da sich Skripte (z.B. shell−Skripte) unter Linux austricksen lassen (der Inhalt ist kurz nach dem Start vom Benutzer austauschbar), ignoriert der Betriebssystemkern das s−bit für Skripte wohlweislich. (Ausnahme: »perl«−Skripte).

28. "swap"

Da das physikalisch vorhandene RAM ("Arbeitsspeicher", "memory") zum gleichzeitigen Betrieb vieler Programe selten ausreicht, bietet Linux die Möglichkeit, spezielle Bereiche der Festplatte wie zusätzliches RAM zu behandeln, diese nennt man "swap−devices".

28.1 Bezeichnungen

"virtual memory" das durch swap−devices bereitgestellte "RAM"

"swapping": vollständiges Auslagern eines Programme in das "virtual memory" (diese Technik findet heutzutage fast keine Verwendung mehr, der Begriff wird aber trotzdem benutzt)

"paging": seitenweises Auslagern von unbenutzten Programmteilen (in Blöcken zu je 4kb)

28.2 Philosophie der Speicherverwaltung

- "virtual memory" ist aufgrund der Mechanik der Festplatten sehr viel langsamer als RAM (> Faktor 1000), es ist also immer nur eine Notlösung.

- Das RAM soll immer für irgendeinen Zweck verwendet werden, ungenutztes ("freies") RAM ist Geldverschwendung.
- Ein kleiner Teil des RAM ("idle memory") muß allerdings für Situation freigehalten werden, in denen kein paging möglich ist (ca. 140 kbyte).

- Ungenutztes RAM wird automatisch zur vorübergehenden Beschleunigung von Festplattenoperationen (Speicher, Lesen) eingesetzt, d.h. wiederholt angeforderte Dateien werden im RAM bereitgehalten.
- Diese Funktion ("buffering") wird sofort stark eingeschränkt, sobald RAM von Prozessen benötigt wird.

- Wird ein Programm zwei− oder mehrmals gestartet, so wird die bereits im Speicher befindliche Kopie (oder die vorhandenen Teile) zum Starten des Programms benutzt ("cache").
- "shared libraries" ermöglichen den gemeinsamen ("shared") Zugriff verschiedener Programme auf die gleichen Funktions−Bibliotheken ("libraries").

28.3 Statistiken zur Speicherbenutzung

- »free« und »vmstat« zeigen den "freien" und benutzten Speicher an.
- Genauere Statistiken erhält man auf der Textkonsole durch Drücken von <Strg−Rollen> oder <Shift−Rollen>.

	total	used	free	shared	buffers	cached
Mem:	15104	14876	228	10508	628	6148
Swap:	20124	3848	16276			

28.4 Erzeugung von "virtual memory"

- Da die Daten von den äußerten Bereichen (=erste Partionen) einer Festplatte am schnellsten gelesen werden können, sollte man dort auch die "swap−devices" plazieren.
- Als "swap−devices" eignen sich leere Dateien oder leere Festplattenpartitionen mit einer Größe von 40kb − 128 MB (PC)

  Erzeugung von 8 MB "virtual memory":

```
-----8<----- hier schneiden -----8<-----
        # Erzeugen einer 8MB großen, leeren Datei
dd if=/dev/zero of=/var/swapfile bs=1k count=8192
mkswap   /var/swapfile  # "Formatierung"
swapon   /var/swapfile  # Einschalten
-----8<----- hier schneiden -----8<-----
```

- "swap−devices" können in gewissen Grenzen im laufenden Betrieb ein− und ausgeschaltet werden.
  Bsp.: »swapoff /var/swapfle«

- Für alle "swap−devices", die in der Datei ‘/etc/fstab’ aufgeführt sind, erfolgt beim booten eine automatische Aktivierung.

- Als Größe sollte man ca. die doppelte Größe des RAM wählen.

28.5 swap−Verhalten

- Für spezielle Einsatzzwecke kann es sinnvoll sein, die Voreinstellungen bzgl. des swap−Verhaltens des kernels zu ändern.
- Die gewünschten Einstellungen können dem Boot−Manager LILO übergeben werden; siehe ‘/usr/src/linux/mm/swap.c’. (Wer diese Datei nicht versteht, sollte auch nicht an den Werten herumfummeln.)
- [? Darf die Priorität des kswapd herabgesetzt werden ?]

- Per "RAID" lassen sich "swap−devices" auf verschiedenen Festplatten zu einer schnelleren Einheit zusammenfassen.

29. Die Datei als Prinzip

29.1 Gerätedateien

- Der Zugriff auf Hardware erfolgt (theoretisch) nur über Dateien.

  Bsp.: »cat /dev/mouse«

  [Zeigt die Bewegungen der Maus an. Nur auf Textkonsole und vorher mit »killall gpm« den "gpm" abschalten. Abbruch mit <Strg−c>.
  Experimente mit anderen Gerätedateien (z.B. den Festplatten−Gerätedateien) sollte man tunlichst vermeiden.]

- Diese speziellen Dateien heissen "Gerätedateien" und liegen im Verzeichnis '/dev'; die engl. Bezeichnung ist "devices".
- Vorteile:
  - Zugriffsrechte für Hardwarekompenten
  - einheitliche Form von Zugriffsrechten
- Es wird unterschieden zwischen
  - "character devices": zeichenorientierte Geräte
  - "block devices": blockweise arbeitende Geräte
- Ausnahmen bilden die Netzwerkkarten (z.B. Ethernet, isdn), die nicht über Dateien angesprochen werden
- Von DOS bekannt: LPT1:, COM1:, NUL:
- Erzeugung fehlender Gerätedateien mithilfe des shell−Skripts »/dev/MAKEDEV«
  Bsp.: »cd /dev; ./MAKEDEV tty9«

29.2 Sockets

- "sockets" sind Übergabepunkte für Informationen (sie stellen eine der drei Möglichkeiten dar, wie Prozesse miteinander kommunizieren können).
  Bsp.: '/dev/printer' (siehe 52.)
- Die Verbindung zwischen zwei Übergabepunkten kann innerhalb der eigenen Maschine ("Unix domain sockets") oder über das Internet geschehen ("Internet domain sockets").
- Herumspielen mit sockets: Paket »socket−1.1«

29.3 Named Pipes

- "named pipes" sind die dritte Möglichkeit, mit der Prozesse untereinander kommunzieren können (die "erste" Möglichkeit nennt sich übrigens "shared memory" [gemeinsam genutzter Speicher]).
- Eine "named pipe" nennt man auch "fifo" (first−in−first−out).
  Bsp.: »mkfifo rohr; echo "hallo_welt" > rohr« (wartet)
  »cat rohr«
- Bildlich gesprochen sind "named pipes" fest eingegrabene Röhren (siehe 22.5).

30. Die Dateien im Verzeichnis '/dev'

Nur nicht von den mehr als 700 Dateien in diesem Verzeichnis abschrecken lassen! Immerhin kann man sie mit einfachen Schemata der folgenden drei Seiten leicht überblicken. Das Format der Übersicht ist:

Datei	ggf. Nummerierung	Bemerkungen, Pr*d*ktnamen

30.1 Parallele Schnittstellen (siehe 'Printing−HOWTO')

Datei	ggf. Nummerierung	Bemerkungen
lp×	× = { 0 .. 2 }	1. − 3. "Druckeranschluß"

"lp" steht für "line printer" ("Zeilendrucker")

30.2 Serielle Schnittstellen (siehe 'Serial−HOWTO')

Datei	ggf. Nummerierung	Bemerkungen
ttyS×	× = { 0 .. 3 }	serielle Schnittstellen Nr. 1−4 (unter DOS: COM1 − COM4)
ttyS×	× = { 4 .. 31 }	nur bei zusätzlicher Hardware, den sogenannten "multiport boards"
ttyI×	× = { 0 .. 16 }	Modem−Emulation des ISDN−Treibers
cua×	× = { 0 .. 31 }	wie "ttyS"
cui×	× = { 4 .. 31 }	wie "ttyI"

30.3 Busmäuse (siehe 'Busmouse−HOWTO')

Datei	Bemerkungen
atibm, atimouse	ATI−XL (Grafikkarte)
inportbm, msmouse	Inport (ATI, M*crosoft)
jmouse	????
logibm, logimouse	Logitech
psaux, psmouse	PS2

30.4 Sound (siehe 'Sound−HOWTO')

Datei	ggf. Nummerierung	Bemerkungen
audio		SUN−ähnliche audio−Schnittstelle
pcaudio		dito f. eingeb. Lautsprecher
dsp		"digital sampling device" ("rohe Audiodaten")
pcsp		dito f. eingeb. Lautsprecher
midi0×	× = { 0 .. 3 }	MIDI−Schnittstelle
mixer		Mixer
pcmixer		dito f. eingeb. Lautsprecher
patmgr×	× = { 0,1 }	Patch−Manager
sequencer		direkter Zugriff auf MIDI, FM, und GUS
sndstat		Daten zur vorhandenen (bzw. erkannten) Hardware

30.5 Terminals (siehe 20.)

```
tty                              Kontrollterminal eines Prozesses
tty×     × = {p0..pf,..,z0..zf}  Terminal−Anschlüsse aus der Sicht
              A0..Af,..,E0..Ef}  des Benutzers
pty×     × = { p0 .. Ef }        Terminals aus Sicht der Prozesse
```

30.6 Konsolen (siehe 17.2)

```
tty×     × = { 0 .. 63 }         virtuelle Konsolen
vcs×     × = { 0 .. 63 }         Inhalt von "tty×"
vcsa×    × = { 0 .. 63 }           dito inkl. Bildschirmattribute
```

30.7 Diskettenlaufwerke (engl. "floppy disk", kurz "fd")

```
fd×      × = { 0 .. 8 }          generisches Gerät, automatische
                                   Erkennung der Diskettenkapazität
fd0CompaQ                        für CompaQ−Laufwerke
fd0cbm1581                       für Commodore 1581−Disketten
fd0D×    × =  360 .. 1040        DD−Disketten
fd0E×    × = 2880 .. 3840        ED−Disketten
fd0H×    × = 1440 .. 1920        HD−Disketten
  | \   |
  |  \  ‘− Kapazität der Diskette
  |   ‘−−− Diskettentyp
  ‘−−−−−−− Nr. des Laufwerks (0=a: 1=b: ...)
```

30.8 Bandlaufwerke mit Diskettenanschluß (siehe ‘Ftape−HOWTO’)

```
rft×     × = { 0 .. 3 }          Bandlaufwerk mit Rückspulen
nrft×    × = { 0 .. 3 }          dito ohne automatisches Rückspulen
```

"rft" steht für "rewinding floppy tape"
"nrft" steht für "not rewinding floppy tape"

30.9 (E)IDE Festplatten (engl. "hard disks", kurz "hd")

```
hda                              gesamte erste IDE Festplatte
hda×     × = { 1 .. 16 }         Partitionen der ersten IDE
  | \                            Festplatte
  |  ‘− Nr. der Partition
  ‘−−−− Nr. der Festplatte (a = erste, b = zweite, ...)
```

30.10 Sonderfunktionen

```
full                             ????
kmem                             Abbild des
log                              Socket f. »syslogd«
mem                              ????
null                             Schwarzes Loch, schluckt alles
ram                              ????
xconsole                         ????
zero                             Liefert unbegrenzt viele Nullen
```

30.11 CD−ROMs mit herstellerspezifischen Schnittstellen (siehe ‘CDROM−HOWTO’)

```
aztcd0                           Aztech CD268A, Orchid CD−3110,
                                   Okano/Wearnes CDD110, Conrad TXC
cdu31a                           Sony CDU31A/CDU33A
cdu535                           SONY CDU−535/531
cm206cd                          CM206 Philips/LMS
gscd0                            Goldstar R420
isp16                            ISP16, MAD16, Mozart
lmscd                            LMS/Philips CD205/225/202
mcd                              Mitsumi
mcdx×    × = { 0,1 }             Mitsumi, multisession−fähig
optcd0                           Optics Storage 8000 AT
sbpcd×   × = { 0 .. 3 }          Sound−Blaster OHNE  IDE/ATAPI, also
                                   die meisten Matsushita, Panasonic,
                                   Creative, Longshine, Kotobuki und
                                   TEAC Laufwerke
sjcd                             SANYO CDR−H94A
sonycd                           ????
```

30.12 SCSI Geräte

```
st×      × = { 0,1 }             SCSI−Bandlaufwerk/Streamer
nst×     × = { 0,1 }             dito, ohne autom. Zurückspulen

sra                              SCSI−CDROM mit ID 0
sga                              generisches SCSI−Gerät mit ID 0
sda                              gesamte SCSI Festplatte mit ID 0
sda×     × = { 1 .. 15 }         Partitionen der SCSI Festplatte
  | \                              mit ID 0
  |  ‘− Nr. der Partition
  ‘−−−− SCSI−ID (a = 0, b = 1,  ..,  h = 7)
```

"st" steht für "scsi tape"
"sr" steht für "scsi rom"
"sg" steht für "scsi generic"
"sd" steht für "scsi disk"

30.13 Verweise im Verzeichnis ‘/dev’

```
:0             −> console            siehe 17.2
cdrom          −> mcdx0              (Beispiel)
console        −> tty0               siehe 20.2
core           −> /proc/kcore        Abbild des kernels
fd             −> /proc/self/fd      Verz. mit "file descriptor"
ftape          −> rft0               Floppy tape
kcore          −> /proc/kcore        Abbild des kernels
modem          −> cua2               (Beispiel)
mouse          −> ttyS0              (Beispiel)
sbpcd          −> sbpcd0             SoundBlaster−CDROM
stdin          −> fd/0               Standard−Eingabe
stdout         −> fd/1               Standard−Ausgabe
stderr         −> fd/2               Kanal für Fehlermeldungen
nftape         −> nrft0              Floppy tape
```

31. Organisation des Dateibaums

Glossar:
"File System": Dateisystem (auch: Dateibaum, Verzeichnisbaum)

31.1 Grundlegendes

- keine Laufwerksbuchstaben, stattdessen ein einziger, aus den verschiedenen Massenspeichern zusammengesetzter Verzeichnisbaum
- Auch Daten anderer Betriebssysteme können in diesen Baum integriert werden (Prinzip des virtuellen Dateissystems).
- drei große, ineinandergeschachtelte Hierarchien:

```
/usr/local      selbst installierte Software (veraltet)
/usr            Lieferumfang der Distribution
/               zum Booten unbedingt benötigte Programme
```

(jeder Baum jeweils ohne den vorherigen Ast)
- drei weitere wichtige Äste:

```
/usr/X11R6      (Standardaufteilung)
/usr/lib        (kunterbuntes Durcheinander)
/var            (kein ‘/usr/var’ oder ‘/usr/local/var’ !)
```

31.2 Der "Linux File System Standard"

- Der "File Hierarchy Standard" (FHS) enthält Richtlinien zum Aufbau des Dateibaums; erhältlich per von

  ftp://tsx−11.mit.edu/pub/Linux/docs/linux−standards/fsstnd

- Der Standard richtet sich nur an Personen, die Software "für Linux" zusammenpacken und auf ftp−Servern hinterlegen.

31.3 "Disk space ... the final frontier"

Verzeichnis	Platzbedarf min.	besser	Bemerkung
/	20	30	abhängig von der Distribution
/usr	300	500	je größer, je besser
/tmp	20	40	unkritisch
/home	20	100	pro Benutzer mind. 10 MB
/var	20	60	TeX−fonts, e−mail, Druckaufträge
swap	20	40	siehe 28.
gesamt:	400	770	

- Die verschiedenen Äste lassen sich auch auf mehrere Festplatten oder Partitionen verteilen.

Beispiel für einen Verzeichnisbaum (unvollständig):

```
/                Wurzel des Verzeichnisbaums
|                (engl. "root−directory")
|
bin/             Programme, von Benutzern ausführbar
boot/            Kopien der Boot−Konfiguration, Betriebssystemkern
dev/             Gerätedateien ("devices")
etc/             Konfigurationsdateien ("et cetera")
home/            Privatverzeichnisse der Benutzer
lib/             Bibliotheken mit Programmroutinen
|  |             und sonstige Datensammlungen
|  modules/      ladbare Treiber des Betriebssystemkerns
|
lost+found/      Reste verlorengegangener Dateien
mnt/             vorübergehend/testweise montierte Speichermedien
proc/            Informationen über den Zustand des Systems
root/            Privatverzeichnis der Systemverwalterin (root)
sbin/            Programme, nur von root ausführbar
tmp/             Verzeichnis für temporäre Dateien
usr/
|  |
|  X11R6/        X Window System, Version 11 Ausgabe 6
|  bin/          Programme
|  doc/          Dokumentation
|  lib/
|  include/      Variablendeklarationen f. C−Bibliotheken
|  local/
|  |    |
|  |    bin/
|  |    doc/
|  |    lib/
|  |    include/
|  |    sbin/
|  |    src/
|  |
|  man/          elektronische Handbücher ("manual pages")
|  src/          Quelltexte ("sources")
|     |
|     Linux/     Quelltexte zu Linux (Betriebssystemkern)
|
var/             veränderliche Dateien ("variable")
   |
   local/
   lock/         Reservierung/Blockierung von Gerätedateien
   log/          Logbücher und admistrative Informationen
   run/
   spool/        Druckaufträge, e−mail, News, TeXfonts, ...
   |    |
   |    cron/    Konfiguration für die Zeitschaltuhren
   |    lpd/     Druckaufträge
   |    mail/    Postfächer für e−mail
   |
   tmp/
```

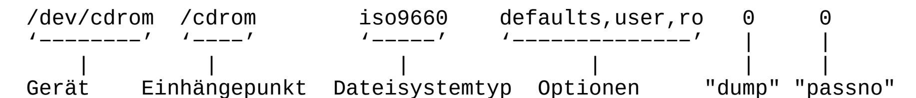
32. Montage des Dateibaums

32.1 Montagewerkzeug

- Der Befehl »mount« setzt den Verzeichnisbaum (siehe 31.1) aus den Datenträgern oder Teilen davon zusammen.
- Das Ausführen des »mount«−Befehls wird kurz als "mounten" bezeichnet.

```
        Bsp:    »mount −t msdos /dev/fd0 /mnt«
                          |     |        |
                         Typ   Gerät    "wohin" (»mount point«,
                                                 Einhängepunkt)
```

- Konfiguration des »mount«−Befehls in ‘/etc/fstab’ (siehe »man 5 fstab«); diese Datei wird während des Bootvorgans automatisch "ausgeführt". Beispiel:



```
/dev/cdrom  /cdrom       iso9660    defaults,user,ro   0     0
‘−−−−−−−−’  ‘−−−−’       ‘−−−−−’    ‘−−−−−−−−−−−−−−’   |     |
    |         |             |              |           |     |
Gerät    Einhängepunkt  Dateisystemtyp  Optionen     "dump" "passno"
```

- Zur Bedeutung von "dump" und "passno" siehe ????? und ????.
- Die Reihenfolge in ‘/etc/fstab’ ist entscheidend; der Ast ‘/usr’ muß gemountet sein, bevor ‘/usr/local’ gemountet werden kann.
- Das "mounten" kann gewöhnlichen Benutzern mit der Option ’user’ erlaubt werden (evtl. ’user,noauto’).

- Das Kommando »umount« ist die Säge für den Dateibaum; es arbeitet vor dem Abschalten des Computers den Inhalt der Datei ‘/etc/fstab’ in umgekehrter Reihenfolge ab.
  Bsp.: »umount /cdrom«

32.2 Das "virtual file system" (VFS)

- Ein Dateisystem (engl.: "file system") ist die Art und Weise, mit der man Daten auf Festplatten, CD−ROMs, usw. organisiert, bzw. speichert.
- Das VFS ist das einzige Dateisystem für Linux − zumindest erscheint dies dem Benutzer so.
- Der Zugriff auf tatsächlich existierende Dateisystemtypen (für Linux, DOS, OS/2) geschieht über Treiber (im Betriebssystemkern).

- Nicht vorhandene Eigenschaften werden durch das VFS vorgegaukelt (z.B. Eigentümer und Zugriffsrechte für DOS−Dateien).
  Bsp.: »ls −l /mnt /floppy /cdrom«

- mac/amiga−Datenträger können von der PC−Hardware nicht verarbeitet werden, da hilft auch kein VFS!

32.3 Einige Dateisysteme, die Linux verarbeiten kann

Typ	für Zugriff auf	Typ	für Zugriff auf
ext2	Linux (standard)	hpfs	OS2 (readonly)
iso9660	CD−ROMs	msdos	DOS
minix	Minix	vfat	win95
nfs	Netzwerk	ncp	novell−Server
sysv	System−V	smb	WfW−shares
ufs	Sun/Nextstep/FreeBSD	cfs	verschlüsseltes FS

- Das "mounten" der Dateisystemtypen ’ncp’ und ’smb’ erfordert z.Zt. ein spezielles »mount«−Kommando.
  Bsp.: »ncpmount −S NovServ /mnt«

32.4 Das "extended−2" Dateisystem

- Ein Datenträger (oder Bereiche davon) erhält durch den Befehl »mke2fs« das Format des extended−2 Dateisystems und kann dann entsprechend "gemountet" werden. Der Vorgang wird als "Formatieren" bezeichnet.
  Bsp.: »mke2fs −c /dev/fd0«

- Die Betriebsparameter für das extended−2 Dateisystem können als Parameter bei der Formatierung angegeben werden oder nachträglich mit dem Befehl »tune2fs«.
  Bsp.: »tune2fs −m 10 −gid li−adm /dev/fd0«

  (In diesem Beispiel ist 10% des Platzes für die Gruppe "li−adm" reserviert.)

- Fehler auf dem Dateisystem können mit dem Programm »fsck« (hoffentlich) behoben werden.
  Bsp.: »fsck −M /dev/fd0«

32.5 "extended−2" Dateiattribute (Linux spezifisch)

- »chattr« zum Setzen der erweiterten Attribute
- »lsattr« zum Anzeigen derselbigen

Kürzel	Attribut	Bedeutung
−		Attribut nicht gesetzt
a	append	Daten können an Datei nur angehängt werden
c	compress	Daten werden automatisch komprimiert
d	no−dump	Datei wird beim "dumpen" ignoriert
i	immutable	Datei unveränderbar (selbst für root)
s	secure−delete	Löschen überschreibt Datei mit Nullen
S	synchronous	wie Option ’sync’ des »mount«−Befehls
u	undelete	Löschen kann rückgängig gemacht werden

33. Speichermedien

Warnung:
auswechselbare Speichermedien vor dem Wechseln unbedingt mit dem Befehl »umount« demontieren, ansonsten droht Datenverlust!

33.1 Diskettenlaufwerke

- Der Umgang mit DOS−Disketten ist unter 8.2 beschrieben.
- Das Paket ‘fdutils−4.2’ enthält Programme zum Konfigurieren der Diskettenlaufwerke, sowie zum Formatieren von Disketten. Nicht zuletzt die Dokumentation der exotischen Formate ist interessant.

- Formatieren von DOS−Disketten mit den Befehlsfolgen
  »fdformat /dev/fd0; mformat a:«
  oder
  »superformat /dev/fd0«

33.2 Festplatten ((E)IDE oder SCSI)

- Linux kann Festplatten beliebiger Grösse ansprechen, ein Eintrag im BIOS−Setup ist NICHT notwendig. Beim Booten werden auch älteste Festplatten erkannt (ansonsten: siehe Bootprompt−HOWTO).
- Die Festplatte wird zur Aufnahme bestimmter Äste des Verzeichnisbaums (oder gar zur Aufnahme anderer Betriebssysteme; *schauder*) in Bereiche unterteilt, diese nennt man Partitionen.

- Die Partitionierung der Festplatte wird mit dem Programm »fdisk« (menügeführt: »cfdisk«) vorgenommen
  Bsp.: »fdisk −l«

- Eine Partition, die die Wurzel des Verzeichnisbaums ‘/’ enhält, wird als "root−partition" bezeichnet.
- Die "root−partition" und der "kernel" müssen innerhalb der ersten 1024 Cylinder auf der 1. oder 2. EIDE−Platte angelegt werden. (Bei SCSI−Festplatten ist dies egal.)
- »hdparm«: Betriebsparameter für die Festplatte abfragen/einstellen (deutliche Geschwindigkeitssteigerung möglich; aber Vorsicht!)

```
Bsp.'e: »hdparm −i /dev/hda«
        »hdparm −m 8 −u 1 −a 1  /dev/hda«  (Vorsicht!)
        »hdparm −S 100 /dev/hda« (Motor nach 1500 Sekunden
                                  Leerlauf abschalten)
```

33.3 (E)IDE

- "IDE" war jahrelang DIE Technik für Festplatten−Controller in PCs und wurde von "EIDE" ("enhanced IDE") abgelöst.
- EIDE unterstützt bis zu 4 Geräte pro Controller und erlaubt höhere Datentransferraten als IDE.
- Neben Festplatten kann man mittlerweile auch (Atapi−)CD−ROMs und (Atapi−)Bandlaufwerke an den EIDE−Controller anschliessen.

33.4 CD−ROMs

- Zum Betrieb von CD−ROMs stehen folgende Techniken zur Verfügung:
  - ATAPI ("EIDE") CD−ROMs oder SCSI CD−ROMs
  - herstellerabhängige Controller

- Nähere Informatationen zu den herstellerabhängigen Techniken findet man im Verzeichnis ‘/usr/src/linux/Documentation/cdrom’.
- Ansonsten gibt es nicht Spezielles zu CD−ROMs zu sagen.

33.5 SCSI

- "SCSI" steht für "Small Computer Systems Interface" und ist eine Schnittstelle zum Anschluß von speziellen
  - Festplatten
  - CD−ROM Laufwerken
  - Bandlaufwerken
  - DATs und Scannern (beide werden über die "generic−devices" angesprochen)

- Pro SCSI−Controller können bis zu 7 solcher Geräte angeschlossen werden; jedes erhält eine eindeutige Nummer; das 8. Gerät (und dessen Nummer) ist der SCSI−Controller.
- Das SCSI−Verbindungskabel ist 50−polig, die maximale Länge des Kabels beträgt 3 m (gemessen vom ersten zum letzten Gerät).
- An beiden Enden des Kabels müssen sogenannte Endwiderstände angebracht sein; dies kann mit externen Steckern geschehen oder per Aktivierung des im letzten Gerät eingebauten Widerstandes.

- SCSI−Festplatten sind NICHT grundsätzlich schneller als IDE−Festplatten, sondern eher etwas teuerer.
- Alle Informationen (Ausnahme: Preis) über eine SCSI−Platte anzeigen: »scsiinfo«
  Bsp.: »scsiinfo −a /dev/sda«

33.6 Toaster

- Die Herstellung eigener CD−ROMs erfordert das Paket »mkisofs−2.0« und eine Änderung des kernels (siehe CD−Write mini−HOWTO).
- Das Verfahren besteht aus zwei Schritten, dem Erzeugen eines 1:1 Abbilds der CD und dem eigentlichen "Brennen".

```
Bsp.:   »mkisofs  −r  −o CD_Abbild  /home/linux
         cdwrite  /dev/sgf  < CD_Abbild «
```

  (Sicherstellen, daß ‘/dev/sgf’ der CD−Schreiber ist! Ansonsten erfolgt "Formatierung" der Festplatte.)

- Vor dem "Brennen" kann man die so erzeugte Datei per "loopback" als CD mounten und testen.
- Der Schreibvorgang darf von _keinem_ anderen Prozeß unterbrochen werden; ansonsten ist die CD unbrauchbar. Es sollten also nur die allernotwendigsten Programme gestartet werden und das Netz abgeschaltet sein (evtl. neuen "run−level" anlegen [siehe ????])

34. Backup

[Habe kein DAT/Tape, daher bleibt diese Seite vorerst leer.]

[Habe kein DAT/Tape, daher bleibt diese Seite vorerst leer.]

35. Netzwerk und Internet

Warnung:
Mit dem Anschluß an das Internet (oder allgemein an ein Netzwerk) setzt man sich potentiellen Bösewichten aus.
Bevor man "loslegt" muß man unbedingt alle accounts mit passwords schützen und einen Blick in die Dateien '/etc/inetd.conf' (siehe 40.3) und '/etc/X11/Xaccess' (siehe 50.5) werfen.

35.1 Das Internet

- Das Internet ist ein weltweites Netzwerk von Computern, in dem jeder Computer mit jedem anderen verbunden ist.
- Das Übertragungsprotokoll heißt "TCP/IP" ("transmission control protocol / internet protocol").

- Alle Teilnehmer am Internet sind gleichberechtigt; das Angebot von Informationen und Dienstleistungen beruht auf Gegenseitigkeit (und nicht auf Gebühren).
- Die Entwicklung des Internet wird von keiner zentralen Organisation oder Firma gesteuert, sondern die Weiterentwicklung obliegt den Benutzern.

  Die Parallelen zu Linux sind also unübersehbar und der Erfolg von beiden gründet sich nicht zuletzt auf obigen Prinzipien − abseits von betriebswissenschaftlichen Denken oder Firmenpolitik.

35.2 Unterschiede zu anderen Medien

- Die Inhalte werden von den Benutzern selbst bestimmt.
- Eine Zensur fand bis vor kurzem nicht statt.

  An deutschen Kiosken wird auf den Titelseiten (d.h. in aller Öffentlichkeit) mehr nacktes Fleisch gezeigt, als im Internet.
  Dieselben Sender, die in den Abendnachrichten noch über "Pornografie im Internet" berichten, strahlen wenige Stunden später selber "Erotikfilme" aus.
  Man kann es als "Doppelmoral" bezeichnen oder als "Angst vor der Konkurenz Internet" werten.

- Der Reichtum des Internet ist von den meisten Menschen nahezu alleine aus dem Spaß an der eigenen Kreativität erschaffen worden, nicht durch bezahlte Arbeit einer Redaktion.
  Wer vom Internet ein optimales Medium zum Konsumieren von bunten Bildern und anderen aufregenden Sachen erwarten, wird daher bitter entäuscht sein.
  Ohne eigenes Zutun wird man nicht ständig unterhalten (wie man das etwa vom Fernsehen gewohnt ist).
  Ebensowenig eignet sich das Internet als Ersatz für die herkömmlichen Informationsquellen (Bücher, Filme, CD−ROMs), sondern höchstens als Ergänzung.
  Vor allem ist das Internet nämlich ein Kommunikationsmedium.

35.3 Linux und Internet

- Ein Linux−PC läßt sich problemlos an das Internet anschliessen; viele Server im Internet sind mit Linux realisiert.
- Linux ist mithilfe des Internet entwickelt worden.

35.4 Wozu braucht man einen Internet−Zugang?

- Um mit anderen Internet−Teilnehmern in direkten Kontakt zu treten (z.B. um e−mail auszutauschen oder um am usenet teilzunehmen).
- Um selbst Informationen im Internet angzubieten.
- Um ganz vorne bei der Entwicklung von Linux dabei zu sein (aktiv oder passiv).

35.5 Alternativen zum Internet−Zugang

- Viele der natürlichen Reichtümer des Internet sind mittlerweile auf preisgünstigen CDs verfügbar
  - Abzüge der wichtigsten FTP−Server (sunsite/tsx−11)
  - tausende interessanter WWW−Seiten
  - Archive der newsgroups (auf CD aber relativ uninteressant; rechtlich umstritten)
- Man sollte auf Aktualität Wert legen (und sich nicht von falschen Behauptungen irreführen lassen, nur die alten Versionen der Programme liefen stabil).
- CDs sind schneller als jedes Modem ...

35.3 Besondere Internet−Anbieter (nicht unbedingt billig!)

- "Individual Network e.V." (IN) Dachverband bundesweiter Betreibergemeinschaften, die Privatpersonen einen Internetzugang anbieten (starke regionale Ausrichtung, ca. DM 20−40/Monat)

  Individual Network e. V., Scheideweg 65, D−26121 Oldenburg
  IN−Info@Individual.NET (e−mail Adresse für Informationen)

- "Verein zur Förderung der privat betriebenen Datenkommunikation e.V." (Vz* oder Sub−Netz)

  ????

- "Verein zur Förderung eines Deutschen Forschungsnetzes e.V." (DFN−Verein), Zusammenschluss der Universitätsnetze; auch für Privatpersonen zugänglich (ab ca. DM 40/Monat)

  DFN−Verein, Pariser Straße 44, 10707 Berlin
  http://www.shuttle.de/ (Anschluß für Privatpersonen)

36. Adressen

36.1 IP-Adressen

- Jede Maschine im Internet ist über (mindestens) eine eindeutige Nummer ansprechbar. Diese Nummer nennt sich "IP-Adresse" und hat das Format

  a.b.c.d (a, b, c und d im Bereich von 1 bis 254)

- Genauer: jede IP-Adresse ist einem Netzwerkanschluß zugeordnet (siehe 38.1)

Anschluß	max. Anzahl	Repräsentiert
lo		Loopback-Netz (Software-Trick)
eth×	×={ 0..4 }	Ethernetkarte
ppp×	×={ 0..16 }	Modem (PPP-Protokoll)

36.2 Organisation der IP-Adressen

- Man spricht von Class-A, -B, -C Netzwerken; die 254*254*254*254 = 4294967296 möglichen IP-Adressen wurden wie folgt aufgeteilt:

Bereich	Typ	Netze	Maschinen je Netz
1.b.c.d - 127.b.c.d	A	127	16387064
128.1.c.d - 191.254.c.d	B	16002	64516
192.1.1.d - 223.254.254.c	C	1999996	254
224.b.c.d - 254.b.c.d	für experimentelle Nutzung reserviert		

- Bei obiger Aufteilung handelt es sich um einen organisatorischen Schritt der Internet-Planer (es gibt nichts daran zu verstehen).
- Durch den Boom des Internet sind Class-A und -B Netze aufgebraucht.

36.3 Spezielle IP-Adressen

- Der gesamte Adressbereich "127.b.c.d" ist dem loopback-Netz zugeordnet, dabei handelt es sich um ein per Software simuliertes Netz.
- Einzige Maschine auf diesem loopback-Netz ist die eigene; evtl. mit verschiedenen Namen. Die Adresse 127.0.0.1 bezieht sich stets auf die eigene Maschine; sie kann zu keinen anderen Zwecken gebracht werden, als zur Definition des "das sind wir selbst".

- Die Adresse "a.b.c.0" ist für das lokale Netz reserviert.
- Die Adresse "a.b.c.255" ist für Nachrichten "an alle" reserviert.

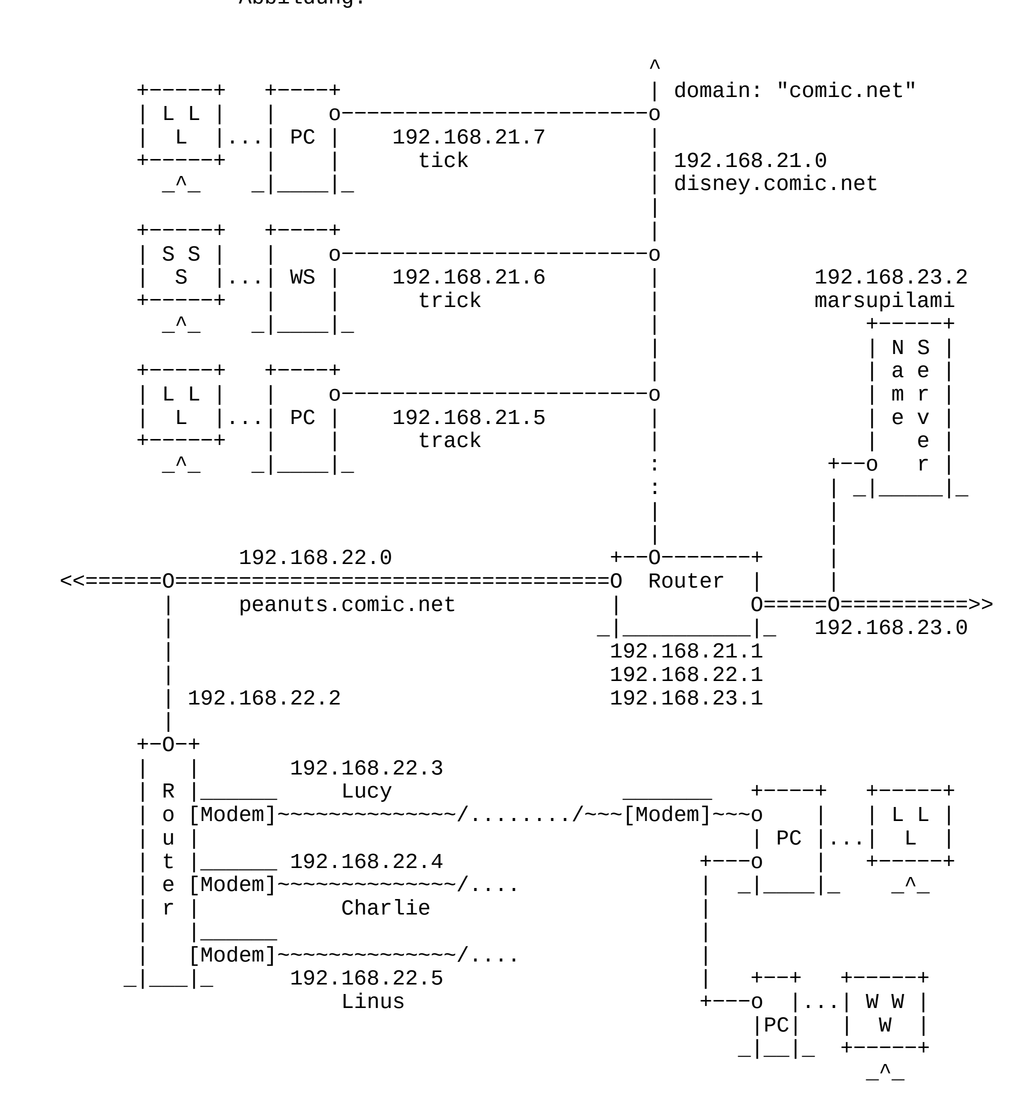
- Folgende Adreßbereiche sind zum Aufbau privater, nicht mit dem Internet verbundener Netzwerke freigegeben:

  10.b.c.d 172.16.c.d 192.168.c.d

  Diese Adressen werden vom Internet nicht verarbeitet!
- Auf Hardware-Adressen wird hier nicht eingegangen.

Beispiel eines Netzwerks

Bemerkungen: Die verwendeten IP-Adressen sind im Internet ungültig. Alle Beispiele der nächsten Seiten beziehen sich auf diese Abbildung.



37. Namen im Internet

37.3 Namen für Nummern

- Weil sich niemand die Nummern merken will, gibt man jeder Maschine neben der IP−Nummer noch eine symbolische Adresse, den "full qualified domain name" (FQDN).
- Der FQDN wird durch Punkte in Bestandteile gegliedert, die über den Namen und die Zugehörigkeit des Rechners Auskunft geben.

  Bsp.: "Euklid.MI.Uni−Koeln.DE" (Mathematisches Institut der Universität zu Köln in Deutschland)
- Übersicht der gebräuchlichen Kürzel

Kürzel	Bedeutung	Kürzel	Bedeutung
ac	academic	com	Firma ("commercial")
de	Deutschland	edu	Ausbildungseinrichtung
net	Netzwerk	org	Organisation

- Das erste Wort des FQDN ist der sogenannte "hostname", der hintere Teil des FQDN ist der "domainname".
- Während der "domainname" vorgegeben ist, hat man bei der Taufe des Rechners freie Wahl. Neben Namen, die in Zusammenhang mit der domain stehen (s.o.) sind außerdem Comichelden, Getränke, berühmte Personen, usw. sehr beliebt.
- Mehrere Namen pro Maschine sind möglich, so wählt man als zweiten Namen häufig den Verwendungszweck einer Maschine. (siehe ????)

Zweck	Protokoll	Beispiel	Bescheibung
archie	????	archie.th−darstadt.de	siehe ????
ftp	ftp	ftp.uni−koeln.de	siehe 11.3
mail	smtp	mail.uni−koeln.de	siehe 42.4
news	nntp	news.uni−koeln.de	siehe 44.2
www	http	www.uni−koeln.de	World Wide Web

37.2 "nameserver"

- "nameserver" kennen die zu den symbolischen Adressen gehörenden IP−Adressen und umgekehrt.
- Da Linux−intern nur mit den numerischen IP−Adressen gearbeitet werden kann, ist bei Verwendung der symbolischen Namen eine nameserver notwendig.
- Abfrage der nameserver mit »nslookup« (keine man−page, dafür ein ’help’−Befehl auf dem nslookup−prompt).

  Bsp.: »nslookup ftp.funet.fi«
- "DNS" ist die Abkürzung für das "domain name system" und bezeichnet das System der nameserver.
- Das Programm »named« verwandelt einen Linux−PC in einen nameserver. Meist wird man nur einen lokalen cache einrichten wollen (siehe caching−named mini−HOWTO).

37.3 Warum überhaupt IP−Adressen?

- IP−Adressen können wegen ihres einfachen Aufbaus von der Hardware schnell analysiert und danach angenommen/ignoriert/weitergeleitet werden.
  Symbolische Adressen zu analysieren würde bei grossen Datenmengen unvertretbar viel Rechenzeit verschlingen (bzw. nahezu unmöglich sein).
- Um ohne IP−Adressen die Bestimmung der Route vom Absender zum Empfänger mit vertretbarem Aufwand möglich zu machen, müßte man die domainnamen nach strengen Regeln vergeben (z.B. "jeder Rechnername in Deutschland endet mit ’de’").
  Die strengen Regeln hat man stattdessen in die IP−Adressen gesteckt und gewinnt dadurch Flexibilität bei den Namen.

37.4 Konfigurationsdateien

- ‘/etc/hostname’: hostname der eigenen Maschine
- ‘/etc/resolv.conf’ kann enthalten
  - Internet−Gebietsname [Schlüsselwort "domain"]
  - IP−Adressen der nameserver (max. 3 Alternativen; werden in der gegebenen Reihenfolge befragt) [Schlüsselwort "nameserver"]
  - Suchpfad für kurze Maschinennamen [Schüsselwort "search"]

```
-----8<-----    Beispiel    -----8<-----
domain   disney.comic.net
nameserver   192.168.23.2
nameserver   192.168.26.172
search   peanuts.comic.net franquin.comic.net
-----8<----- hier schneiden -----8<-----
```

- ‘/etc/hosts’: Namen und Adressen wichtiger Maschinen; zur Umgehung oder als lokale (!) Ergänzung des nameservers

```
-----8<-----    Beispiel    -----8<-----
127.0.0.1       localhost tick.comic.net tick
192.168.23.2    huba hubahuba huba.comic.net
192.168.22.2    lucy  lucy.comic.net
-----8<----- hier schneiden -----8<-----
```

- ‘/etc/host.conf’ legt fest:
  - Reihenfolge: erst ‘/etc/hosts’ oder erst DNS ("bind") befragen?
  - Sind mehrere Namen für die eigenen Maschine erlaubt?
- ‘/etc/networks’: Nummern aller lokalen Netzwerke (ähnlich wie ‘/etc/hosts’ für Netzwerkadressen)

38. Routen auf der Datenautobahn

38.1 Netzwerkschnittstellen

- Der Datenverkehr findet zwischen den Netzwerkschnittstellen der Internet−Rechner statt.
- Ein Linux−PC kann mehrere Netzwerkschnittstellen besitzen und als Vermittlungsstelle zwischen verschiedenen Netzen dienen.
- Der Name einer Netzwerkschnittstelle hängt von der verwendeten Netzwerktechnik ab (nicht etwa vom Hersteller der Einsteckkarte).

Schnittstelle	max. Anzahl	Repräsentiert
arc×	×={ 0..4 }	ArcNet
eth×	×={ 0..4 }	Ethernet
imbtr×	×={ 0..2 }	Token Ring
isdn×	×={ 0..16 }	ISDN
lo		Loopback−Netz (Software−Trick)
plip×	×={ 0..2 }	Druckeranschluß
ppp×	×={ 0..16 }	Modem (PPP−Protokoll)
slip×	×={ 0..16 }	Modem (SLIP−Protokoll)

- Die Zuweisung von IP−Adresse zu Netzwerkschnittstellen wird mit dem Befehl »ifconfig« vorgenommen.
  Bei Vernetzung mit Ethernet ("Standleitung") wird dieser Befehl beim Booten vom shell−Skript ‘/etc/init.d/network’ ausgeführt; bei Wählverbinungen via Modem führen die Programme »pppd« oder »dip« den Aufruf des Befehls automatisch durch.

38.2 Lokale Netze (entfällt bei PPP/SLIP−Nutzung)

- Bei der Planung eines lokalen Netzes hat man 2 Ziele:
  - direkter Datenverkehr zwischen allen lokalen Maschinen ohne Umweg über den router (unnötige Belastung!)
  - Begrenzung der "Reichweite" der Datenpakete auf das kleinste mögliche Teil−Netz (aus Sicherheitsgründen und zum Schutz anderer Teil−Netze vor unnötiger Belastung)
- Die Einstellungen zum Umsetzen dieser Ziele sind:

Bezeichnung	Beschreibung
subnet−mask	Bitmaske zur Feststellung, ob eine Adresse zum eigenen Teil−Netz gehört oder nicht
network−address	Adresse des eigenen Teil−Netzes

```
address 192.168.22.3    11000000.10101000.00010110.00000011
netmask 255.255.255.0   11111111.11111111.11111111.00000000
        ==========log. AND ================================
                        11000000.10101000.00010110.00000000
network 192.168.22.0    11000000.10101000.00010110.00000000
```

In diesem Beispiel gehört das Datenpaket mit der Adresse "192.168.22.3" zum eigenen Teil−Netz. (Alle 254 möglichen Adressen der Form "192.168.22.XXX" gehören dazu.)

38.3 "router"

- Die Weiterleitung/Zustellung der Datenpakete im Internet überläßt man speziellen Vermittlungstellen, den sogenannten "routern". [Diese router befolgen insbesondere die unter 38.2 erklärten Regeln.]
- »traceroute« dient zur Verfolgung des kompletten Weges, den Datenpakete zu einem bestimmten Empfänger einschlagen.

  Bsp.: »traceroute nic.funet.fi«

- U.U. müssen die Pakete bis zum Ziel über mehrere router geleitet werden; darauf hat man aber keinen Einfluß.
- »route« zeigt an, wie der Betriebssystemkern Datenpakete an die verschiedenen Empfänger verschickt.

- Die Bezeichnungen "router" und "gateway" werden meist synonym verwendet, wobei ein "gateway" eigentlich grundverschiedene Netze miteinander verbindet.

38.4 Konfigurationsdatei ‘/etc/init.d/network’ (‘/etc/rc.d/inet1’)

```
# Konfiguration für "lucy.comic.net"

        # An der loopback−Schnittstelle "lo" nehmen wir Pakete
        # für die Adresse "127.0.0.1" entgegen.

ifconfig  lo  127.0.0.1

        # Alle Pakete für die Maschine "127.0.0.1" schicken wir an
        # die loopback−Schnittstelle ... letztendlich also an uns
        # selbst ... daher auch der Name "loopnet".

route  add  −host 127.0.0.1  lo


        # An der Ethernet−Schnittstelle "eth0" nehmen wir Pakete
        # für die Adresse "192.168.22.2" (lucy) entgegen.

ifconfig  eth0       192.168.22.2 \
          broadcast  192.168.22.255 \
          netmask    255.255.255.0

        # Pakete für Maschinen im gleichen Teil−Netz sind bei
        # "eth0" garantiert gut aufgehoben.

route  add  −net 192.168.22.0  eth0

        # Der ganze Rest wird an den router geschickt ... zum Glück
        # gehört der Router zum gleichen Teil−Netz, d.h. er ist
        # über den oben definierten Weg erreichbar.
        # Ansonsten müßte man noch einen eigenen Weg zum Router
        # definieren!

route  add  default  gw 192.168.22.1  eth0
```

39. Realisierung der Netzanbindung

39.1 Ethernet

Bei Vernetzung mittels Ethernet ist pro Rechner eine "Ethernetkarte" (Einsteckkarte oder vergleichbar) erforderlich. Es gibt verschiedene Techniken, die im folgenden beschrieben werden.
Für Privatpersonen ist ein auf Ethernet basierender Internetanschluß derzeit zu teuer, ein lokales Netz kann allerdings interessant sein.

39.1.1 "thick ethernet"

- Diese Technik wird heute fast nicht mehr verwendet.

39.1.2 "thin ethernet"

- Vorteile: schnell, preiswert, einfach
- Nachteile: primitive Technik, etwas empfindlich

- Das Netzwerkkabel ist abgeschirmtes Koaxialkabel (wie beim Fernseher), Verbindungsstücke und –kupplungen sind in BNC–Technik ausgeführt.
- Die Einsteckkarten werden mit T–förmigen Zwischenstücken an das Netzwerkkabel angeschlossen.
- Die Enden MÜSSEN mit Endwiderständen abgeschlossen werden.

```
[=T==========T==========T====/ .. /======T=] <- 50 Ohm Endwiderstand
 _Ø_        _Ø_        _Ø_              _Ø_
=   |      =   |      =   |            =   |
=   |      =   |      =   |            =   | <- Ethernet-Karten
=   |      =   |      =   |            =   |
|___|      |___|      |___|            |___|
```

39.1.3 "twisted pair"

[Entfällt aus Zeitmangel.]

39.2 Anbindung via Modem

- Linux leitet die Datenpakete der Netzwerkschnittstellen "ppp0" und "slip0" auf die entsprechende serielle Schnittstelle um (dabei werden die Daten noch verpackt und verschnürt ...).
- Ein "Modem" empfängt Daten von der seriellen Schnittstelle und verschickt sie über die Telefonleitung (oder Klingeldraht).

- Folgendes shell–Skript findet evtl. die Gerätedatei, an der das eigene Modem angeschlossen ist.

```
-----8<----- hier schneiden -----8<-----
for i in 0 1 2; do
  if (chat -v -t 5 "" ATZ "" \\d "" ATZ \
               OK "" > /dev/ttyS$i < /dev/ttyS$i)
    then; echo "Modem on /dev/ttyS$i (= COM$[$i+1])"
    else; echo "No Modem found on ttyS$i"; fi; done
-----8<----- hier schneiden -----8<-----
```

- Die Konfiguration der seriellen Schnittstellen erfolgt mit dem Befehl »setserial«.
  Bsp.: »setserial /dev/ttyS2 spd_hi« (57600 bps)
  »setserial /dev/ttyS2 spd_vhi« (115200 bps)
  »setserial /dev/ttyS2 irq 5«

- Obige Befehle findet man in der Datei '/etc/rc.boot/0setserial'.
- Die notwendige Software wird beschrieben unter der URL

  http://www.mi.uni-koeln.de/linux/ppp

39.3 ISDN

[Habe kein ISDN, halte den Platz aber vorsorglich frei.]

40. Server−Programme

40.1 Server und Dienstleistungen

- "Server" sind Daemonen, die Daten oder Dienstleistungen zur Verfügung stellen. Meist werden auch die Maschinen auf denen diese Daemonen laufen als "Server" bezeichnet.
- Nicht für jede Dienstleistung muß ein "echtes" Serverprogramm vorhanden sein.
  Bsp.: »grep "/bin" /etc/inetd.conf«

- Die Datei ‘/etc/services’ enthält eine Liste aller dem System bekannten Dienstleistungen.
  Nicht in ‘/etc/services’ verzeichnete Dienstleistungen sollten nicht verfügbar sein.
- Die Kommandosprache vieler Daemonen ist für den Menschen verständlich.
  Bsp.: »telnet localhost smtp
  help
  quit«

40.2 Ports

- Alle wichtigen Dienstleistungen tragen eine international vereinbarte Kennummer, die sogenannte Portnummer (kurz Port).
- Die Zuordung von Namen zu Ports geschieht in der Datei ‘/etc/services’. Dort vereinbart können die Namen in Programmen und Konfigurationsdateien (siehe 40.3) benutzt werden.
  Bsp.: »telnet localhost 13«
  »telnet localhost daytime«

- Portnummern im Bereich von 1−1023 können nur von Daemonen belegt werden, die die Systemverwalterin root gestartet hat. Dies ist eine Sicherheitsmaßnahme.

40.3 Der Internet−Super−Daemon »inetd«

- Es gibt mehr als 50 Daemonen mit Server−Funktion für Linux. Würden diese Programme stets gleichzeitig laufen, bräuchte man dafür alleine mehrere 10 MB Arbeitsspeicher (RAM).
- Daher werden die meisten Daemonen nur bei Bedarf von einem zentralen Programm gestartet. Dieses Programm ist der Daemon »inetd«.
- Konfiguration durch ‘/etc/inetd.conf’; Änderungen an dieser Datei werden durch »killall −HUP inetd« wirksam.
- Mindestens "netstat" und "systat" sollten durch Voranstellen des Kommentarzeichen ´#´ deaktiviert werden. Oder zumindest sollte der Zugriff auf wenige Maschinen beschränkt sein.

  Bsp.: »telnet localhost netstat«
  »telnet localhost systat«

- Eine simple Zugriffsbeschränkung wird durch den »tcpd« gewährleistet, der vor dem Aufruf der eigentlichen Daemonen die Gegenseite der Verbindung genau überprüft (siehe 40.6)

40.4 Standalone−Daemonen

- Einige Daemonen brauchen zum Start einige Sekundenbruchteile Zeit, um alle Konfigurationsdateien zu lesen und sich zu initialisieren.
- Werden solche Daemonen intensiv genutzt (man denke an den http−Daemon für WWW−Anfragen), so ist der Umweg über den »inetd« (startet den daemon für jede Anfrage neu) viel zu langsam.
- Stattdessen müssen solche Daemonen als eigenständige Prozesse laufen, man bezeichnet die Daemonen dann als "standalone"; z.B. »sendmail« und »httpd«.
- Alle Daemonen, deren Programmnamen mit »in.« beginnt, müssen vom »inetd« gestartet werden, d.h. können nicht als standalone−daemon betrieben werden.

40.5 RPC

- "RPC" steht für "Remote Procedure Call" und ist eine Ergänzung/ Weiterentwicklung des herkömmlichen, auf ports basierenden Server−Konzepts.
- "RPC−Daemonen" sind meist erkennbar an vorangestelltem "rpc." im Programmnamen (Ausnahmen z.B.: »ypbind« »ypserv«)

40.6 Zugriffsrechte

- »tcpd« und »rpc.portmap« können bestimmten Rechnern den Zugriff verweigern. (siehe »man hosts_access«)

- ‘/etc/hosts.deny’: Welchen Maschinen wird der Zugriff auf welche Dienste verboten?

```
-----8<----- hier schneiden -----8<-----
ALL: PARANOID
portmap ypbind yppasswdd ypserv: ALL
-----8<----- hier schneiden -----8<-----
```

- ‘/etc/hosts.allow’: Welchen Maschinen wird der Zugriff auf bestimmte Dienste erlaubt?

```
-----8<----- hier schneiden -----8<-----
portmap ypbind yppasswdd ypserv: \
        localhost tick trick track
-----8<----- hier schneiden -----8<-----
```

41. Netsurfen

41.1 Das URL−Format

- Speziell für die WWW−Browser wurde das URL−Format entwickelt, das die Adressierung aller Informationen, Dateien und andere Resourcen im Internet in einem einheitlichen Format erlaubt.
- In einer URL−Adresse stecken die folgenden Informationen:
  - Protokoll
  - voller Name der betreffenden Maschine
  - (− evtl. Portnummer [siehe 40.2])
  - Dateiname (+Verzeichnisname) oder password

```
Bsp.:   http://www.uni−paderborn.de/Linux/mdw
         ftp://ftp.uni−koeln.de/doc/linux/crash−kurs
          |    ‘−−−−−−−−−−−−−’‘−−−−−−−−−−−−−−−−−−’
          |           |                  |
  Protokoll     Maschinenname       Verzeichnis
```

41.2 Herkömmliche Programme

- Terminalsitzungen: »telnet«
  Bsp.: »telnet localhost«

- Datentransfer: »ftp«
  Bsp.: »ftp localhost«

41.3 Navigation im Internet

41.4 Berkeley r−utilities

- Die »r−utitlies« erlauben die Arbeit mit anderen Maschinen ohne die Eingabe eines Namen und Geheimwortes.
- Stattdessen erfolgt die Autorisierung über Einträge in den Dateien

Datei	Format einer Zeile
/etc/hosts.equiv	"−Rechnernamen", "+Rechnernamen"
~/.rhosts	"Rechnername", "Rechnername Benutzer"

- Vorteile:
  - passwords werden nicht durch das Netz geschickt
  - bequemer
- Nachteile:
  - Systemverwalter der anderen Maschine erhalten Zugang
  - ein anderer Rechner kann unter falschem Namen Zugang bekommen

- Übersicht

Funktion	Programm	Beispiel
telnet	»rlogin«	»rlogin localhost«
Programm starten	»rsh«	»rsh localhost −− ls −l /«
Dateien kopieren	»rcp«	»rcp v.txt localhost:v.tex«
X11−Programme starten (bauen auf »rsh« auf)	»xon«, »xrsh«	»xon localhost netscape«

41.5 Sicherheit auf allen Wegen: »secure shell«

41.6 Zubehör

- Netz−Radar: »ping«
  Bsp.: »ping localhost«
- Mitschnitt des Datenverkehrs: »tcpdump«
  Bsp.: »tcpdump −i lo«
- »tkined« ist ein universelles Programm zum Beobachten des Netzes mit einer grafischen Oberfläche

42. Elektronische Post

42.1 Grundlegendes

- o Elektronische Post wird als "e−mail" ("electronic mail") bezeichnet. Sie wird zwischen den Rechnern des Internet mithilfe des SMTP−Protokolls zugestellt ("simple mail transport protocol").
- o "emaillieren": mit Email überziehen

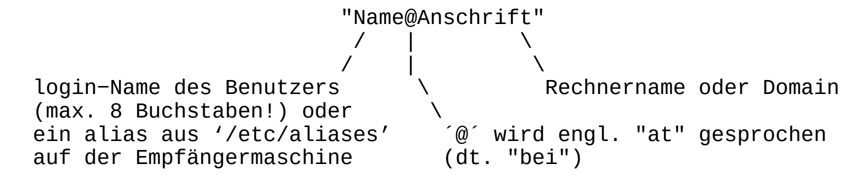
- o Eine e−mail setzt sich aus folgenden Feldern zusammen:

From:	Absender
Date:	Datum ("des Poststempels")
Reply−To:	Rückadresse (wenn ungleich Absender)
To:	an wen ist die e−mail gerichtet
Subject:	worum geht es in der e−mail ("Betreff")
Cc:	Kohledurchschlag ("carbon copy") an ...
Bcc:	Durchschlag ohne Wissen des Empfängers ("blind carbon copy") an ...
Fcc:	Kopie für die eigenen Unterlagen ("file carbon copy") in die Datei ...
Recieved:	Stationen im Internet, über die die e−mail weitergeleitet wurde

- o Zusammen bilden oben genannte Felder den "header" der e−mail.
- o Der eigentliche Text wird als "body" oder "message text" bezeichnet.

42.2 Adressen

- o e−mail−Adressen haben die Form



- o "postmaster" ist der Ansprechpartner bei Problemen mit der e−mail
  Bsp.: "postmaster@localhost"

- o e−mail−Adressen muß man kennen, es gibt keine vollständigen Adressbücher oder eine andere zuverlässige Einrichtung, um an e−mail−Adressen zu kommen.
- o Der X.500−Dienst ist ein kleines Adressbuch für e−mail−Adressen

  Bsp.: »finger truemper@X500.Uni−Koeln.DE«

- o WWW−Interface zum X.500−Dienst unter der URL

  http://www.uni−koeln.de/dienste/x500.html

42.3 Programme zum Lesen und Schreiben von e−mail

Programmname	Eigenschaften
pine	einfach, komfortabel, Speicherfresser
elm	Standard−Programm, unkomfortabel
emacs	"Alt−x rmail"; nur Kaffekochen kann er nicht ...
tkmail	mit graphischer Oberfläche; einfach

- o Sie Signatur aus der Datei ‘~/.signature’ wird von vielen Programmen automatisch an die ausgehende e−mail angehängt.

42.4 Verschicken von e−mail

- o e−mail−Programme reichen die e−mail weiter an

  − die Daemonen »smtpd« oder »sendmail« auf dem eigenen Rechner

  oder

  − an einen "SMTP−Server" (unkonventionell: "mail−Server"), d.h. einem Rechner, auf dem diese Daemonen laufen

- o Eine Empfangsbestätigung erfolgt i.d.R. NICHT (und man sollte dies auch tunlichst unterlassen, z.B. wegen der Mailinglisten).
- o Ist eine e−mail überhaupt nicht zustellbar, so wird sie mir einer kurzen Notiz zurück an den Absender geschickt.
- o Ist eine e−mail vorübergehend nicht zustellbar (z.B. Netzausfall), so wird sie im Verzeichnis ‘/var/spool/mqueue’ ("mail queue") gespeichert und in regelmässigen Abständen wird die Zustellung erneut versucht (max. 5 Tage lang).

42.5 Empfang und Zustellung von e−mail

- o Die Daemonen »smtpd« oder »sendmail« nehmen mail aus dem Internet entgegen und stellen diese nach folgenden Regeln zu:

  − Zunächst wird geprüft, ob ein Eintrag in der Datei ‘/etc/aliases’ nähere Angaben zum Empfänger enthält.

  − Falls ein Nachsendeauftrag in der Datei ‘~/.forward’ des Empfängers (= ´~´) besteht, wird die e−mail an die darin enthaltene e−mail−Adresse geschickt.

  − Ansonsten wird die e−mail im Verzeichnis ‘/var/spool/mail’ zur Abholung durch den Benutzer hinterlegt, und zwar tragen die "Briefkasten"−Dateien die Namen der Benutzer.

- o Gängige e−mail−Programme gucken automatisch in diese Briefkästen.
- o Mit dem Programm »popclient« kann man die e−mail aus den Briefkästen anderer Rechner herausholen (z.B. um sie "offline" zuhause oder unter DOS zu lesen).
  Bsp.: »popclient −3 −o mailbox localhost«

- o Eine Weiterentwicklung des POP3−Verfahrens ist die "imap"−Technik.

42.6 Tips und Tricks

42.7 e−mail Filter

42.8 Zubehör

- Briefkastenflaggen: »xbiff«, »xpbiff«, »mailleds«
- Automatische e−mail "Beantworter":
  - »vacation« schickt dem Absender die Nachricht aus der Datei ‘~/.vacation.msg’ zu; z.B. bei (im voraus bekannter,) mehrtägiger Abwesenheit
  - »BCR MailHandler« schickt auf Anforderung bestimmte Dateien zu
- Verschiedenes
  - »splitdigest« erzeugt aus "digest" wieder einzelne mails
  - »from« Kurzübersicht der Neueingänge
  - »pgp« Briefsiegel und Verschlüsselung
  - »super−cite« Zitate im emacs

43. Mailinglisten

43.1 Grundlegendes

- "Mailinglisten" verteilen e−mail automatisch an eine grössere Anzahl von Personen.
- Sie dienen der Kommunikation via e−mail, ohne die Teilnehmer mit der Verwaltung der Adressen der anderen Teilnehmer zu belasten.
- Zu jeder Mailingliste existieren zwei e−mail Adressen:
  - eine "Steueradresse" bei der man den Bezug der Mailingliste abonnieren und stornieren kann
  - die "Verteileradresse" schickt alle e−mail, die an sie gerichtet wird, an die Abonnementen weiter
- Nicht alle Mailinglisten sind öffentlich zugänglich.
- Es gibt gerade für Linux einige Mailinglisten mit einigen hundert Beiträgen pro Woche; Vorsicht!
- Um diesem Problem zubegegnen und die wertvollen Informationen trotzdem einer breiten Masse zugänglich zu machen, gibt es »digest« (ein subjektives "best of" einer Mailingliste)
- Viele öffentliche Mailinglisten werden archiviert und die Archive sind per WWW lesbar, daher ist es nicht unbedingt notwendig, eine Mailingliste zu abonnieren.

43.2 »majordomo«

- Die meisten mailinglisten sind mit dem Programm »majordomo« realisiert.
- Die Befehle für die majordomo−Steueradresse müssen formlos im e−mail Text ("message body") stehen, NICHT im Betreff ("subject"). Andere Programme machen es z.T. genau umgekehrt.
- Einige der majordomo−Befehle:

»help«	schickt eine e−mail zurück, in der die Bedienung und alle Befehle des majordomo erklärt werden
»lists«	listet die Namen aller Mailinglisten auf, die von dem betreffenden Rechner verwaltet werden
»subscribe«	gefolgt vom Namen einer Mailingliste richtet ein Abonnement für diese ein
»unsubscribe«	gefolgt vom Namen einer Mailingliste löscht das Abonnement für diese

43.3 Linux−spezifische Mailinglisten

- »Yggdrasil Computing« organisiert (lies: bezahlt) für die Linux−Gemeinde die Umleitung aller wichtigen Mailinglisten in entsprechende newsgroups. Dadurch ist ein gefahrloses und kostengünstiges Mitlesen für alle möglich.
  Informationen unter
  ????
- Man sollte als normaler Benutzer die newsgroup der "linux.*"−Hierarchie dem Abonnement der mailinglisten vorziehen.

44. Das "usenet"

44.1 "usenet" und "newsgroups"

- Eine "newsgroup" beinhaltet e−mail−artige Diskussionsbeiträgen zu einem Oberthema; diese werden als "Artikel" bezeichnet.
- I.a. kann sich jeder an der Diskussion beteiligen, d.h. Artikel lesen und verfassen.
- In "moderierten" newsgroups bestimmt ein Moderator, welche Artikel in der newsgroup erscheinen (z.B. um Mißbrauch zu verhindern).
- Die Gesamtheit aller Rechner, die das Lesen und Schreiben von Artikeln erlauben, nennt man "usenet".
- Der Lebensnerv von Linux war und ist das usenet.

44.2 Organisation der newsgroups

- newsgroups sind hierarchisch nach Themen geordnet.
- Unter−Hierarchien werden durch Punkte voneinander getrennt
  Bsp.: »comp.os.linux.announce«
- Die Entstehung und Auflösung erfolgt durch demokratische Wahl der usenet−Teilnehmer (siehe newsgroup »de.newusers«).

44.3 Infrastruktur des usenet

- Die Artikel werden auf festen Wegen (Universitäten, Organisationen) durch das usenet gereicht, entweder per NNTP ("network news transport protocol") oder in großen Paketen via uucp ("unix−to−unix copy")
- Die Laufzeit eines Artikels kann einige Tage betragen.

44.4 newsreader

- "newsreader" sind Programme zum Lesen und Verfassen von Artikeln
- "threaded newsreader": fassen Frage und Antworten unter einer Überschrift zusammen (unbedingt empfehlenswert).
- Umgebungsvariablen (siehe 22.4)zur Konfiguration von newsreadern:
  - "NNTPSERVER" hostname des NNTP−Servers
  - "ORGANIZATION" Organisation (z.B. "home sweet home")
- Einige newsreader:
  - »xskim«: kostensparend und ohne Aufwand news via Modem lesen
  - »tin«: nicht besonders aufregend, aber einfach und funktionell
  - »Gnus«: der Overkill, nur für Mutige; Infos unter der URL http://www.ifi.uio.no/~larsi
- Ohne ständige Netzanbindung ("offline") am usenet teilnehmen: »skim«, »slurp«, »newsflash«, »postit« (zum Selberbasteln). Eine fertiges Rezept findet man unter der URL:

  ftp://pople.pc.uni−koeln.de/pub/news.ps

44.5 usenet−Jargon

CfD	"call for votes" (Aufruf zur Stimmabgabe)
crossposting	Artikel, die an mehrere newsgroups gleichzeitig gerichtet sind; i.d.R. ist dies absolut UNERWÜNSCHT
flame	Artikel, die die Diskussion kräftig anheizen (meist unsachlich und verletzend)
follow−up	Artikel mit direktem Bezug auf einen vorhergehenden (entspricht dem "reply" der e−mail)
nettiquette	Knigge für das Internet, Netz−Benehmen
off−topic	themenfremd
posting	Artikel in den news
RfD	"request for discussion" (Aufruf zur Diskussion über die Neueinrichtung einer newsgroup)

44.6 Archive des usenets

- Bevor man im usenet eine Frage stellt, sollte man einen Blick in die Archive werfen; evtl. ist die Frage dort samt zugehöriger Antwort verzeichnet
- Die "FAQs" der einzelnen newsgroups findet man auf verschiedenen FTP−Servern (in Verzeichnisse mit Namen der newsgroups einsortiert)
  ftp://ftp.uni−paderborn.de/FAQ (Deutschland)
- "DejaNews" kann per WWW nach Stichworten durchsucht werden:
  http://www.dejanews.com/forms/dnsetfilter.html

  [dies ist nicht die Startseite, sondern die Seite mit dem Filter: "*Linux*" eingeben]

44.6 Einige newsgroups und Hierarchien

alt.*	alternate newsgroups
comp.*	Computer−bezogene Themen
comp.lang.*	Programmiersprachen
comp.os.*	Betriebssysteme ["operating systems"]
comp.os.linux.*	Linux−spezifische Themen
comp.os.linux.advocacy	Linux gegen den Rest der Welt
comp.os.linux.announce	Ankündigungen (moderiert)
comp.os.linux.answers	Dokumentation (moderiert)
comp.unix.*	
linux.*	(siehe 43.3)
misc.*	miscellaneous (Verschiedenes)
news.*	
rec.*	recreation (Hobby und Freizeit)
sci.*	science (Wissenschaft)
soc.*	society (Gesellschaft)
talk.*	talk (Unterhaltung)
de.*	deutschsprachige newsgroups, Wiederholung der englischen Original−Struktur
de.comp.os.Linux	
de.newusers	
de.talk.jokes	Witze
de.test	Zum Ausprobieren und Testen der newssoftware

45. MIME & Co.

46. Kommunikation mit anderen Benutzern

Hinweis:
um sich vor einer Notizenflut anderer Benutzer zu schützen, kann man mit dem Kommando »mesg n« deren Empfang abstellen (betrifft »write«− und »talk«−ähnliche Programme)

46.1 Benachrichtigen

- »write«: Nachricht direkt auf Bildschrim des anderen schreiben
- »wall«: dito, aber an alle
- »rusers«, »rwho«, »finger«: wer ist wo zu finden

46.2 Palaver in Echtzeit: »talk«, »ntalk«, »ytalk«

- »ytalk« ist die komfortabelste Version (<Esc> aktiviert Menü)
  Bsp.: »mesg y; ytalk $LOGNAME«
- mit SUN−Maschinen gibt es hinsichtlich talk oft Probleme, aber auch sonst ist der talk−Dienst häufig "außer Betrieb"

46.3 Online−Diskussionsforen: »irc«

- »irc« steht für "Internet Relay Chat" (aber: »man ircII«)
- in den verschiedenen Diskussionsforen wird LIVE diskutiert; der Stil ist eher chaotisch
  Bsp.: »irc −c Linux irc.informatik.rwth−aachen.de«
- Befehle beginnen mit dem Zeichen ‘/’ (slash)
  Bsp.: ‘/quit’, ‘/help’

46.4 Fax

- die Programmpakete »mgetty+sendfax«, »efax«, »hylafax« erlauben das Verschicken und Empfangen von Faxen via (Fax−)Modem
- gegenüber e−mail hat Fax unter Linux nur eine geringe Bedeutung

46.5 Spiele

Spiele für Linux basieren i.d.R. auf einem Netzwerk und erlauben mindestens das austauschen von kurzen Nachrichten; auf möglichst realistische Grafiken wird keinen Wert gelegt. Eine Auswahl:

- »mud«: textbasiertes Rollenspiel, meist im Stil von
  »Dungeons&Dragons«, Informationen in den newsgroups
  »rec.games.mud.announce« und »de.alt.mud«
- »xpilot«: Raumschiff−Kriegsspiel in 2D−Welt, Team−Modus möglich;
  Information via »telnet xpilot.cs.uit.no 4400«
- Schach: »xboard« (Oberfläche f. »gnuchess«); Kontaktaufnahme zum
  Schach−Server mit »xboard −ics True −icshost chess.lm.com«

46.6 4 goldene Regeln für den Umgang mit anderen Netzteilnehmern

- Schreib nichts, was Du dem anderen nicht auch ins Gesicht sagen würdest; am anderen Ende der Leitung sitzt ein Mensch wie Du und ich, kein gefühlsloser Computer.
- Wenn Dir jemand Beleidigungen an den Kopf wirft: mach Dir nichts draus, manche Menschen verlieren vor dem Bildschirm total das Gefühl für den anderen Menschen (also: nicht direkt totschlagen!).
- Laß Dir unaufgefordert zugesandten Werbemüll und anderen Ramsch nicht gefallen.
- Ohne präzise Informationen KANN Dir nicht geholfen werden − oder Du zahlst wie bei kommerziellen hotlines kräftig dafür, daß man Dir die Würmer einzeln aus der Nase zieht: "Ist Ihr Computer eingeschaltet?" kommt nicht von ungefähr.
- Zieh Dich warm an, sprech nicht mit fremden Leuten und räum’ endlich Dein Zimmer auf.

46.7 Häufig vorkommende Abkürzungen

AFAIK	as far as I know	soweit ich weiss
ASAP	as soon as possible	so schnell wie möglich
BTW	by the way	übrigens; wo wir gerade dabei sind
cu	see you	man sieht sich
DTRT	do the right thing	das Richtige tun
FYI	for your information	zur Kenntnisnahme
gr&d	grinning and ducking	grinst und duckt sich
IMHO	in my humble opinion	meiner bescheidenen Meinung nach
	in my honest opinion	meiner ehrlichen Meinung nach
LOL	laughing out loud	lautes Lachen
MORF	man or female?	Mann oder Frau?
OIC	oh, I see	oh, ich verstehe
ROTFL	rolling on the floor, laughing	die Person wälzt sich vor Lachen am Boden
RSN	real soon now	schon bald; in Kürze
RTFM	"read the fucking manual"	ein gängiger Hinweis, sich mit dem Handbuch auseinanderzusetzen
THX	thanks	Danke
YABOME	yet another bug of mail exchange	ein weiterer Fehler in diesem mail−Programm

weitere unter der URL:
????

46.8 Smilies (Seite um 90° im Uhrzeigersinn drehen)

:−)	Standard−Smiley
;−)	zwinkender Smiley
:−/	zweifelnder Smiley
:−(	unglücklicher oder verärgerter Smiley
:−<	trauriger Smiley
:−0	diesem Smiley steht der Mund vor Schreck offen
(−:	Benutzer ist Linkshänder

weitere unter der URL:
????
im Buchhandel: »Smileys«, Sanderson/Doughtery, O’Reilly−Verlag

47. Sicherheitsaspekte

49. X−Windows−System

49.1 Grundlegende Eigenschaften von X11

- "X11" ist der Unterbau der grafischen Benutzeroberfläche von Linux. Es wurde in den 80er Jahren am "Massachusetts Institute of Technology" (MIT) entwickelt und wegen seiner enormen Bedeutung dem dafür gegründeten "X Consortium" übereignet.
- Das reine "X11" stellt lediglich grafische Informationen dar, die es von anderen Programmen erhält.
- Die Verwaltung der Fenster, deren Dekoration und Aussehen wird vom Windowmanager (Fenstermanager) gesteuert.

```
-----8<----- hier schneiden -----8<-----

/usr/bin/X11/X  >  X-log  2>&1  &

# Ein grauer Bildschirm erscheint.
# Zurückwechseln mit <Strg-Alt-F1>

xterm  -ls -sb  -display :0.0  &

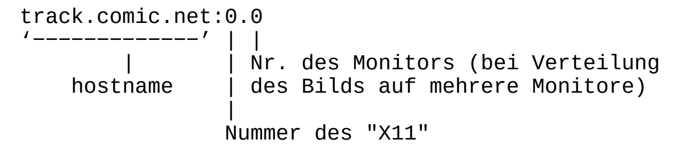
# Scheinbar passiert nichts. Die virtuellen Konsolen nach X
# absuchen und die Maus in das neue Fenster platzieren.

/usr/bin/X11/fvwm  &

-----8<----- hier schneiden -----8<-----
```

49.2 Netzwerkfähigkeiten

- "X11" ist netzwerkfähig. Es vergibt an den Monitor eine Adresse, über den er im gesamten Netzwerk (z.B. Internet) angesprochen werden kann.
- Das Format der Monitor−Adresse ist



- Wird ´X´ mehrmals gestartet so erhalten die daraus entstehenden Oberflächen anstelle der Endung ´:0.0´ die Endungen ´:1.0´, ´:2.0´ usw.
  Bsp.: »/usr/bin/X11/X :1 «
- Anhand der Umgebungsvariable "DISPLAY" wird Programmen mitgeteilt, wie die Adresse des Monitors lautet, den sie benutzen sollen.
  Bsp.: »echo $DISPLAY«
- Die Option '−display' überschreibt den Wert von "DISPLAY".
  Bsp.: »xterm −display localhost:0.0«

49.3 Zugriffskontrolle

- Die Netzwerkfähigkeiten von X11 machen eine Zugriffskontrolle unbedingt notwendig. Dabei geht es NICHT NUR um das unberechtigte Öffnen von Fenstern, sondern vielmehr muß verhindert werden, daß die Vorgänge in den Fenstern ABGEHÖRT werden können (man denke an die Eingabe eines passwords). Das Abhören ist nämlich mit primitivsten Mitteln möglich, siehe »man xev«.
- Unter Linux regeln z.Zt. zwei Verfahren den Zugriff auf die eigene X−Sitzung, die sich etwa so einordnen lassen:
  - Erlaubnis für alle Programme von einer bestimmten Maschine − ohne Unterscheidung der Benutzer
  - Erlaubnis für alle Programme, die einen bestimmten Ausweis vorlegen können
- Das Programm »xhost« legt fest, von welchen Maschinen jeder Benutzer zugreifen darf.
  Man sollte mit diesem Programm keinerlei Erlaubnis erteilen, auch nicht den Benutzern der eigenen Maschine (s.o.).
  Bsp.: »xhost − « (Verbot für alle Maschinen)
- Der Standard unter Linux ist die Methode "MIT−MAGIC−COOKIE−1", bei der Programme einen Ausweis (üblicherweise in '~/.Xauthority') vorlegen müssen. Die Datei, die diesen Ausweis enthält darf natürlich nur für den Eigentümer les− und schreibbar sein (das Ausführungsrecht muß NICHT gesetzt werden).
- Dieser Ausweis kann z.B. mit »xauth« weitergegeben werden (siehe auch »xon« oder »xrsh«).

49.4 Windowmanager

- "virtual−Windowmanager" vervielfachen die nutzbare Oberfläche; der max. sichtbaren Ausschnitt ist natürlich durch den Monitor festgelegt.
- Unter Linux wird meist der »fvwm«−Windowmanager eingesetzt.

Übersicht Windowmangager

Name	Konfigurationsdateien	Beschreibung
»fvwm«	'/etc/X11/fvwm/system.fvwmrc' '~/.fvwmrc'	"Free Virtual Window Manager"; vielseitig
»mwm«		"Motif Window Manager" (kommerzielles Programm)
»olvwm«	'/usr/openwin/lib' '~/.openwin-init'	"Open Look Virtual Window Manager"; stilvol
»twm«	'/etc/X11/twm/system.twmrc' '~/.twmrc'	"Tab Window Manager"; sehr schlicht

50. Mehr über "X"

50.1 Begriffe

"X−Server": das Programm, das die Grafikbefehle der anderen Programme in ein Bild auf dem Monitor umwandelt.

"root−window": der "Hintergrund".

"cut&paste": markieren und an anderer Stelle einfügen.

50.2 X für Linux

- Es gibt unter Linux mehrere Programme, die die Funktion von "X11" übernehmen; sie entsprechen dem Release 6 (X11R6):
  - "XFree86" vom "XFree86 Project, Inc."
  - "AcceleratedX" von der Firma "X Inside Inc." ist schneller als "XFree86" (noch schneller?!), aber nicht kostenlos

50.3 XFree86, Version 3.1.2

- Um die Hardware möglichst optimal auszunutzen, wurden für die verschiedenen Familien von Grafikkarten (z.B. "S3", "Mach") spezielle Versionen des XFree86 geschrieben.
  Bsp.: »dir /usr/X11R6/bin/X*«
- Trotzdem ist noch eine Konfiguration notwendig, da z.B. der Monitor ebenfalls optimal ausgenutzt werden soll, andererseits nicht überlastet werden darf.
- Zur Erzeugung der Konfigurationsdatei ‘/etc/X11/XF86Config’ kann das Programm »/usr/X11R6/bin/xf86config« benutzt werden.
- XFree86 übernimmt beim Start die Tastenbelegung der Textkonsolen ... mit Ausnahme der Taste <AltGr>. Bei der Konfiguration muß die Frage
  Do you want to enable these bindings for the Alt keys?

  mit ´y´ beantwortet werden, um auch diese zu aktivieren.

Tastenkombination	Funktion
<Strg−Alt−BackSpace>	beendet XFree86 sofort (Notbremse)
<Strg−Alt−Numpad+>	nächsthöhere Auflösung wählen
<Strg−Alt−Numpad−>	nächstniedrige Auflösung wählen
<Strg−Alt−F1>	zurück auf die Textkonsole wechseln

- XFree86 benötigt eine "leere" virtuelle Konsole; ohne weitere Angaben wird die erste "leere" Konsole beschlagnahmt.
  Bsp.: »/usr/X11R6/bin/X vt4«

50.5 Starten von X11

- Wegen der Netzwerkfähigkeiten gibt es mehrere Möglichkeiten zum Starten und Arbeiten mit X11:

Prg.	Konfigurationsdatei	Beschreibung
startx	‘/etc/X11/xinit/xinitrc’	Aufruf nach dem Einloggen an virtueller Textkonsole
openwin	‘/usr/openwin/lib/Xinitrc’	dito; startet "Open Windows"
xdm	‘/etc/X11/xdm/Xsession’	login−Fenster für direkten Zugang zum X11

»X −query localhost« nur X auf der eigenen Maschine, alles andere auf "localhost"

»X −indirect localhost« Liste von Maschinen anfordern, die eine X−Sitzung erlauben

50.6 Der "X−Display Manager"

- »xdm«, der "X−Display Manager", sorgt für einen ständigen Betrieb des XFree86 auf der Konsole und läßt ein login−Fenster erscheinen (kein Anmelden an der Textkonsole erforderlich).
- Konfigurationsdateien des »xdm« im Verzeichnis ‘/etc/X11/xdm’:

Beschreibung	Datei
Namen der Konfigurationsdateien	‘xdm−config’
Aussehen und Eigenschaften des login−Fensters	‘Xresources’
Selbst zu verwaltende X−Sitzungen	‘Xservers’
Welche Maschinen dürfen mit der Option ’−query’ auf uns zugreifen?	‘Xaccess’

- Folgende shell−Skripte verwalten die X−Sitzung

Beschreibung	Datei
Zubehör zum login−Fenster starten	‘Xsetup’
Nach dem Einloggen des Benutzers	‘Xstartup’, ‘GiveConsole’
Start der eigentlichen Sitzung	‘../Xsession’
Nach Beenden der Sitzung	‘Xreset’, ‘TakeConsole’

51. Konfiguration von X11−Anwendungen

X11−Andwendungen lassen sich konfigurieren über

(1) die Kommandozeile
(2) den per »xrdb« geladenen Einstellungen ("X−Resourcen")
(3) Umgebungsvariablen
(4) Konfigurationsdateien im Verzeichnis
‘/usr/lib/X11/app−defaults’

Die Priorität der Einstellungen entspricht obiger Nummerierung.

Die per »xrdb« geladenen Einstellungen werden (nur) für die aktuelle Sitzung in einer X11−internen Datenbank gespeichert. Programme können diese Datenbank auslesen und die Einstellungen übernehmen.

Die Standardkonfigurationen der X11−Programme findet man im Verzeichnis ‘/usr/lib/X11/app−defaults’; diese sollten nicht geändert werden, sondern nur als Beispiel dienen.
Stattdessen kann man die Einstellungen mit Einträgen in den Dateien

‘~/.Xresources’ oder ‘~/.Xdefaults’ (privat)
‘/etc/X11/Xresources’ (global)

überschreiben. Die Dateien werden mit dem Befehl »xrdb« in die X11−interne Xresources−Datenbank geladen.

Bsp.: »xrdb −merge ~/.Xresources«

Beispiele:

```
−−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
*Background: lightgrey
xwpe.font: 10x20
Mosaic*HomeDocument:    http://www.rrz.uni−koeln.de/
Emacs*cursorColor:      red
XDvi.paper:     a4
XDvi*geometry:  800x720
XTerm*ttyModes:   intr ^c  quit ^\\  kill ^u  erase ^?
XTerm.VT100.Translations:  #override \n\
   <Key>Home:      string(0x01) \n\
   <Key>End:       string(0x05) \n\
   <Key>Delete:    string(0x04) \n\
   <Key>BackSpace: string(0x7f) \n\
   Ctrl<Key>Left:  string(0x1b) string("b")\n\
   Ctrl<Key>Right: string(0x1b) string("f")\n
−−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
```

52. Drucken

52.1 Ansteuerung der Hardware

- Der Betriebssystemkern stellt lediglich Gerätedateien zur Verfügung, über die direkt mit dem Drucker kommuniziert werden kann.
  Er hat NICHT die Funktion eines Druckmanagers zum Verwalten und Aufbereiten der Daten.
  Bsp.: »cat /proc/cpuinfo > /dev/lp1«

- Herausfinden des "richtigen" Gerätedatei:
  Bsp.: »for i in /dev/lp*; do tunelp $i; done«

- Konfiguration der Gerätedateien: »tunelp« (tune−lineprinter)
- I.d.R. sind Drucker vom Werk aus auf den Zeichensatz von DOS eingestellt (siehe 8.3). Entweder (falls möglich) legt man einen entsprechenden Mini−Schalter am Drucker um, oder man muß für die Konvertierung von Texten mit »unix2dos« sorgen (im Druckerfilter ergänzen).
- Der Betriebssystemkern muß den Treiber für die parallele Schnittstelle enthalten ("Parallel Printer Support"). Siehe 56.3

52.2 Druckmanager

- Druckmanager übernehmen die folgenden Funktionen

  - Einrichtung von symbolischen Namen für die einzelnen Drucker
  - Entgegenahme der zu druckenden Daten (siehe unten)
  - Zwischenspeicherung der Daten bis zur Bereitschaft des Druckers (man denke an mehrere Druckaufträge)
  - Aufruf des "Druckertreibers" (ein Konvertierungsprogramm für die Pixelsprache des jeweiligen Druckers)
  - Übermittlung der Daten an den Drucker

- Die Konfiguration und die Vergabe von symbolischen Namen wird in der Datei ‘/etc/printcap’ vorgenommen.

```
-----8<----- hier schneiden -----8<-----
octopuss|lpt1|lp|PostScript|DeskJet 3050:\
        :lp=/dev/lp1:\
        :sd=/var/spool/lpd/octopuss:\
        :if=/usr/local/bin/deskjet-treiber:\
        :mx#0:sh:
-----8<----- hier schneiden -----8<-----
```

  In diesem Beispiel erhält der Drucker an ‘/dev/lpt1’ die Namen "octopuss", "lpt1" und "lp".
  Die Druckaufträge für ihn werden im Verzeichnis ‘/var/spool/lpd/octopuss’ zwischengespeichert (engl. "spoolen").
  Das Programm ‘/usr/local/bin/deskjet−treiber’ setzt die Daten in die Pixelsprache des Druckers um.

- Zur Realisierung eines Druckmanagers stehen die Pakete »Berkeley lpd« und »LPRng« (etwas flexiblerer/komplexer) zur Verfügung.
- Der zugehörige Daemon trägt den Namen »lpd« und benötgt ein Netzwerk (das »loopback« reicht bereits).
- Steuerung mit »lpc« ("line printer control").

52.3 Druckerfilter und Druckertreiber

- Ein Druckerfilter ist ein Programm, das vom Druckmanager zur Aufbereitung der zu druckenden Daten aufgerufen wird. Mein persönlicher Favorit ist das Programmpaket "magicfilter" (da einfach zu installieren, transparent und flexibel).

- Druckerfilter erkennen den Typ der Daten anhand charakteristischer Zeichen an bestimmten Positionen (z.B. "GIF" zu Beginn der Datei bei GIF−Bildern):

  - Reiner Text und Daten in der Pixelsprache des Druckers werden nicht weiter verändert und an den Drucker geschickt.
  - Sinnvoll druckbare Daten werden mithilfe verschiedener Programme (z.B. "gunzip", "pbmplus−utilities", "dvips", usw.) in das postscript−Format überführt.
  - Alle anderen Daten werden als "nicht druckbar" zurückgewiesen.

- Das Programm »gs« ("ghostscript") enthält Druckertreiber für die Umsetzung von postscript in das Pixelformat vieler handelsüblicher Drucker. (Nur notwendig, wenn man keinen postscript−fähigen Drucker besitzt.)
- Liste der eincompilierten Druckertreiber: »gs −h | more«

- Ein Druckerfilter führt genau die Befehle aus, die man auch per Hand auf der Kommandozeile geben würde, um eine Datei in die Pixelsprache des eigenen Druckers zu konvertieren. Beispiel:

```
-----8<----- hier schneiden -----8<-----
zcat ckl-3.5.ps.gz  |  gs  -q -dSAFER -dNOPAUSE  -sDEVICE=djet500\
        -sOutputFile=- -  > /dev/lp1
-----8<----- hier schneiden -----8<-----
```

  In diesem Beispiel wird die Datei ‘ckl−3.4.ps.gz’ zunächst decomprimiert (»zcat«) und dann für einen Deskjet−500 Drucker aufbereitet.

52.4 Liste nützlicher Konvertierungsprogramme

[Kommt alles, nur Geduld.]

52.5 Druckaufträge aus der Sicht des Benutzers

Aktion	Prg.	Beispiel
Datei drucken	»lpr«	»lpr −h ckl−3.5.ps«
Warteschlange zeigen	»lpq«	»lpq −l «
Auträge zurückziehen	»lprm«	»lprm −«

o Die Auswahl des Druckers geschieht nach folgenden Regeln:

- Der erste Drucker in ‘/etc/printcap’ ist der Standard−Drucker
- Die Umgebungsvariable "PRINTER" kann einen anderen Drucker als Voreinstellung angeben.
- Mit der Option ’−P’ überschreibt man die ersten beiden Einstellungen, bezieht sich aber nur auf den aktuellen Befehl.
  Bsp.: »lpr −Pkrake ckl−3.5.ps«

o Ist in der Konfigurationsdatei ‘/etc/printcap’ eine Grenze für die maximale Größe eines Druckauftrags angegeben ("mx#1000" o.ä.), so müssen grössere Dateien müssen mit der Option ’−s’ gedruckt werden. In diesem Fall wird im spool−Verzeichnis lediglich ein symbolischer link angelegt, d.h. die Datei darf bis zur Erledigung des Drucks nicht angefaßt werden.
Bsp.: »lpr −s ckl−3.5.ps«

o »mpage«, »atp«, »gslp«, »a2ps«, »x2ps«, »genscript« sparen Papier, indem sie 2 (oder mehr) Seiten Text verkleinert auf eine Seite drucken (im postscript−Format).

52.7 Zugriffsrechte

o Beschränkung der Druckerbenutzung auf eine Benutzergruppe: ":rg=studenten:\" (beschränkt Zugriff auf "studenten").

o Freigabe der Drucker für andere Maschinen im Netz: ‘/etc/hosts.lpd’ und ‘/etc/hosts.equiv’

o Platzmangel, ab dem keine Druckaufträge mehr entgegengenommen werden: ‘/var/spool/lpd/octopus/minfree’

52.6 Drucker im Netz

o Um Drucker anderer Linux−Rechner im Netz anzusprechen, läßt man das Feld für die Gerätedatei in der Datei ‘/etc/printcap’ leer und trägt zusätzlich ein:

```
−−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
        :lp=:\
        :rm=marsupilami:\
        :rp=gutenberg:\
−−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
```

Wobei "marsupilami" der Name des anderen Rechners und "gutenberg" der Name des Druckers am anderen Rechner ist.

53. Linux als Print− und Fileserver

54. Es wird Zeit

54.1 Die Uhrzeit

o Unter Linux gibt es "drei verschiedene Uhrzeiten":

- die der auf der PC−Hauptplatine installierten CMOS−Uhr,
- die Linux−interne Uhrzeit (Systemzeit) und
- die von verschiedenen »Uhren« angezeigte Zeit

```
           »clock −u −w«             »date −s«
 CMOS−Uhr  <−−−−−−−−−−−−−−   Linux    <−−−−−−−−−−−−−−
(Mainboard)               Systemzeit                   Benutzer−Uhr
  (GMT)    −−−−−−−−−−−−−−>   (GMT)    −−−−−−−−−−−−−−>     (MET)
            »clock −u −s«           »date«, »xclock«
```

o Die Option ’−u’ des clock−Befehls bewirkt eine unveränderte Übernahme der CMOS−Zeit als Systemzeit (üblicherweise während des Bootvorgangs).

WARNUNG: bevor man mit "clock" die CMOS−Uhr verstellt, sollte man alle Dateien schliessen.

54.1 Zeitzonen

o Die ganze Welt ist in Zeitzonen eingeteilt; der "Nullpunkt" ist willkürlich auf "Greenwich" (England) festgelegt worden.
o "Greenwich Mean Time" (GMT) ist die sogenannte Normalzeit, aus der sich alle anderen Uhrzeiten berechnen lassen.
o Die Regeln zur Umrechnung befinden sich unter Linux im Verzeichnis ‘/usr/lib/zoneinfo’.
o Zu den Regeln gehört z.B. die automatische Umstellung von Sommer− auf Winterzeit.

o Die Datei ‘/usr/lib/zoneinfo/localtime’ legt die Zeitzone eines Linux−Systems fest; Programme wie »date« und »xclock« berechnen daraus die Uhrzeit. Oftmals ist diese Datei ein link.

o Die Umgebungsvariable "TZ" kann von den Benutzern zur Wahl einer anderen Zeitzone benutzt werden.
Bsp.: »export TZ=Asia/Tokyo; xclock −title Tokio &«
»export TZ=America/New_York; xclock −title "NY" &«

(Die Beispiele setzen voraus, daß der Windowmanager die Titel der xclock−Fenster nicht unterdrückt!)

o Ist Linux das einzige Betriebssystem auf dem Rechner, so setzt man die Systemzeit auf »GMT« (Greenwich Mean Time) und legt einen Verweis von ‘localtime’ auf ‘MET’ an.
o Ansonsten genau umgekehrt (d.h. Regeln werden nicht beachtet).

54.3 Synchronisation der Uhrzeit

o Zum einmaligen Abgleich der Uhrzeit mit anderen Maschinen im Netz kann »netdate« benutzt werden.
Bsp.: »netdate trick tick track«

o Um die Uhrzeit fortlaufend mit allen anderen Maschinen im Netz abzugleichen dient der Daeomon »xntp«.
Dabei kommt es weniger auf die exakte Uhrzeit, als auf eine möglichst einheitliche Zeit aller Maschinen an.

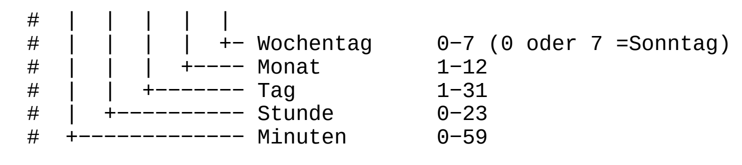
54.4 "cron"

o Der daemon »cron« startet Programme immer wiederkehrend in regelmäßigen Zeitabständen.

o Jeder Benutzer kann mit dem Programm »crontab« eine individuelle Tabelle mit cron−Anweisungen erstellen
Bsp.: »crontab −e«

```
−−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
        # 5:30 Uhr das Modem zurücksetzen
   30 5  *  *  *        modem_reset

        # Erinnerung: jeden Mittwoch den Drucker aussaugen
   0  8  *  *  Wed      mail truemper < Drucker_saugen
```




```
−−−−−8<−−−−− hier schneiden −−−−−8<−−−−−
```

o Dokumentation der Syntax in »man 5 crontab«
o Verursachen die durch »cron« gestarteten Programme eine Ausgabe, so wird diese als mail an den Benutzer geschickt.

54.5 Die Programme »at« und »batch«

o »at« führt Programme einmalig zu einem bestimmten Zeitpunkt aus.
Bsp.: »at −f ftp_get_file.sh 7am Mar 10 «

In diesem Beispiel werden die Kommandos aus der Datei ‘ftp_get_file.sh’ am 10. März um 7:00 Uhr ausgeführt.

o Die Liste aller Aufträge wird mit »atq« angezeigt; einzelne Aufträge können mit »atrm« gelöscht werden.

o »batch« führt Programme aus, sobald der Rechner nicht ausgelastet ist.

55. Deutsche Anpassungen

56. Linux

56.1 Der Betriebssystemkern

- Zur Aufgabe des Betriebssystemkerns siehe 3.2
- Der Betriebssystemkern wird als »kernel« bezeichnet.
- Bezugsquellen für die neusten kernel−Quelltexte unter 12.3.
- Die verschiedenen Versionen des Betriebssystemkerns werden in aufsteigender Reihenfolge nummeriert.
- Kernelversionen unter "2.1" erzeugen i.d.R. schwere Minderwertigkeitsgefühle.

56.2 Begriffe

"kernel−panic": Notbremse des Betriebssystemkerns zur Verhinderung von Folgefehlern. Hält das Linux−System sofort und ohne Vorwarnung an. Selten.

"Code−Freeze": Ankündigung von Linus, in die BETA−Phase zu gehen, um den diesjährigen "stabilen" Kernel herauszugeben. Ruft allgemeine Torschlußpanik hervor, um noch möglichst viele neue Funktionen einzubauen.

56.3 Treiber und Module

- "Treiber" verleihen dem kernel Fähigkeiten wie "Lesen von DOS−Dateisystemen" oder "Ansteuern von ATAPI−CDROMs".

Befehl	Angezeigte Übersicht
»cat /proc/devices«	Geräte (block− und zeichenorientiert)
»cat /proc/filesystems«	Dateisysteme
»cat /proc/net/dev«	Netzwerkschnittstellen

- "Module" sind Treiber, die im laufenden Betrieb in den kernel eingebunden werden können.
- Bei der Erzeugung eines neuen kernels kann man wählen, ob ein Treiber fester Bestandteil des kernels sein soll, oder als Modul zur Verfügung stehen soll.

Programm	Funktion	Beispiel
»lsmod«	Liste aller geladenen Module anzeigen	»lsmod«
»insmod«	Modul in den kernel laden	»insmod lp«
»rmmod«	Modul aus kernel entfernen	»rmmod lp«

- ‘/etc/modules’ listet alle Module auf, die beim booten geladen werden sollen.
- Automatisches Laden von Modulen "bei Bedarf": »kerneld«

56.4 Konfiguration und Erzeugung eines Betriebssystemkerns

- Voraussetzung für die Erzeugung eines lauffähigen Betriebssystem−kerns sind
  - der C−Compiler gcc plus Zubehör und
  - die Quelltexte zum Betriebssystem (z.B. im Verzeichnis ‘/usr/src’).
- Das Auspacken unterscheidet sich nicht von dem anderer "Software" (siehe 12.2).

```
Bsp.:   »cd /usr/src; mv linux linux−old; \
         tar −xzvf  linux−1.3.77.tar.gz«
```

- Auch die "Updates" lassen sich wie gewohnt einspielen (siehe 12.3).

```
Bsp.:   »cd /usr/src; for i in patches/*; do \
         zcat $i | patch −p0«
```

- Die Konfiguration wird im Verzeichnis ‘/usr/src/linux’ mit den Befehlen

  »make config«, »make menuconfig« oder »make xconfig«

  vorgenommen.
- Jede Einstellung kann modifiziert werden und ist genau beschrieben.
- Nach der Konfiguration wird der kernel mit der Befehlsfolge

  »make dep && make clean && make zImage«

  erzeugt. Er steht danach als Datei ‘zImage’ im Verzeichnis ‘/usr/src/linux/arch/i386/boot’ zur Verfügung.
- Die Beschreibung der Installation dieser Datei folgt.
- Gibt man abschliessend den Befehl

  »make modules; make modules_install«,

  so werden die Module erzeugt und im Verzeichnis ‘/lib/modules’ installiert (und werden dort von den modules−Programmen gefunden).

Hier ein Wort von unserem großen Meister:

1.3.x kernels _are_ development kernels, but the more people that use them, the merrier. People should be _aware_ of the fact that it’s a development kernel, but they shouldn’t be discouraged from using them as long as they know what it means:

- if you see a bug, report it. To me, to the mailing lists, and possibly even to the newsgroups. If you’re running an old kernel, people will usually tell you to upgrade to the latest kernel and then try it again, so you might want to do that before being told.

- if you get bitten by a bug, tough luck. I’ll do my very best to get the bug fixed, but the one thing I won’t do is feel sorry for you. In fact, I might ask you to do it all over again, just to get more information. I’m a heartless bastard.

So do not try to discourage people from trying development kernels: just tell them what that implies. Lots of people will then be happier to use the stable kernels, but if everybody is chicken, we won’t get any bugreports or fixes.

Erklärende Übersetzung einiger englischer Vokabeln

Die englische Original–Dokumentation verwendet viele Wörter, die nicht in Wörterbüchern zu finden sind, bzw. die dort nicht mit der im technischen Zusammenhang üblichen Bedeutung zu finden sind.

Die folgende Übersicht soll eine kleine Hilfestellung geben.

Die im Internet gebräuchlichen Akürzungen sind im Abschnitt 46.7 wiedergegeben.

Englisch	dt. Übersetzung mit Erklärung
abort	Programm oder Verbindung abbrechen; "Press ctrl–c to abort"
ambiguous	mehrdeutig
announcement	Ankündigung
argument	Wert (Einstellung), der einem Programm übergeben wird
bracket	eckige Klammer [oder]
branch	Zweig, Alternative
character	Zeichen
comment	Kommentar
compile	Übersetzen; z.B. "C–Programme in ausführbaren Maschinencode compilieren"
concatenate	hintereinanderhängen
consider	betrachten
deadlock	nicht aufhebare Blockierung; etwas hat sich festgefressen
default	Vorgabewert, Standardeinstellung
device	Gerät; der Zugriff auf Geräte erfolgt unter Linux fast ausschließlich über Dateien, die sogenannten Gerätedateien im Verzeichnis '/dev'; meist bezieht sich "device" auf eine dieser Dateien
directory	Verzeichnis
environment	Umgebung; bezieht sich auf die Einstellungen, die einem Programm mit Umgebungsvariablen (enviroment variables) übergeben wird
execute	ausführen (eines Programms oder Befehls)
file	Datei
file descriptor	Dateizugriffsnummer; programminterne Nummer für eine Datei
font	Zeichensatz
forward	weiterleiten (meist im Zusammenhang mit e–mail)
header	Vorspann
host	Maschine, Computer, Rechner
I/O	Ein–/Ausgabe
interface	Schnittstelle zwischen – Mensch und Maschine (z.B. grafische Oberflächen) – Netzwerkkarte und Betriebssystemkern
kernel	Betriebssystemkern; der heilige Teil von Linux
key	Taste
keyboard	Tastatur
library	Zusammenfassung von compilierten C–Routinen zu einer Bibliothek, die dann aus jedem anderen Programm heraus benutzt werden kann
link	Verweis; anstatt eine Datei zu kopieren, kann man auch einen Verweis anlegen; dies spart Platz und erlaubt einige Tricks
load	laden, Beladung, Ausnutzung; in Zusammenhang mit der Auslastung einer Maschine gebraucht
lock	Sperre, Riegel (meist gewollt)
match	Übereinstimmung
mount	montieren, zusammensetzen; stets im Zusammenhang mit dem Verzeichnisbaum, der durch das Kommando »mount« zusammengesetzt wird
network	Computernetz(werk)
news	
OS	Betriebssystem (Operating System)
obsolete	überflüssig, veraltet
overflow	bei einer mathematischen Operation ist ein Ergebnis aufgetreten, daß wegen seiner Größe nicht mehr in die dafür vorgesehene Variable paßt
parenthesis	runde Klammer
path	Pfad; meist als Begriff für den kompletten Verzeichnisnamen zu einer Datei gebraucht
pipe	Röhre, Veranschaulichung der Weiterleitung von Daten von einem zum nächsten Programm; das sogennante "pipe–symbol" ist das Zeichen | (senkr. Strich)
port	
preference	Vorlieben (bei Konfiguration)
press	Aufforderung zum Drücken einer Taste
prompt	Bereitschaftmeldung (oder –zeichen)
proprietary	
quote	Anführungszeichen oder Zitat
range	Bereich; von ... bis
remote	im Sinne von "die Gegenseite der Verbindung"
reply	antworten (auf eine e–mail oder einen Artikel)
resolve	Ermittlung (Auflösung) der numerischen IP–Adresse, die zu einem Maschinennamen gehört
sequence	Sequenz, Aufeinanderfolge
server	Computer, der anderen Computern Dateien oder Dienstleistungen zur Verfügung stellt
socket	
source	meist Begriff für die Programmtexte (Quellcode, Sourcecode) benutzt; "may the source be with you"
space	Leerzeichen, Platz ("disk space, the final frontier")
stack	Stapel
string	Zeichenkette
subject	Thema, Betreff (bei e–mails)
synopsis	zusammenfassende Übersicht
template	Vorlage, Muster, Schablone, Schema
trace	Ablaufverfolgung
type	Tippen Sie ein
underflow	Unterlauf, z.B. "der Unterlauf des Rheins" ;–)
usage	Gebrauch, Gebrauchsanleitung
verbatim	wortwörtlich
wildcard	Joker, meist erfüllt der Stern * diese Funktion, z.B. zur unvollständigen Angabe von Dateinamen